# Magnetic Tape Subroutines For Assembler And Fortran Compiled Programs For The IBM 1130

1130 00.3.003

00.3.003 • Magnetic Tape Subroutines For Assembler And Fortran Compiled Programs For The IBM 1130 • 1130

CONTRIBUTED PROGRAM LIBRARY •

DM1 ONLY

# MAGNETIC TAPE SUBROUTINES FOR ASSEMBLER AND FORTRAN COMPILED PROGRAMS FOR THE IBM 1130

Martin J. Michel
August 31, 1967

Direct Inquiries To:

Mr. Martin J. Michel
C/O Mr. R. W. Allphin
IBM Corporation
180 South Main Street
Providence, R. I. 02901

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for IBM Data Processing Systems. When such announcements occur, users should order a complete new program from the Program Information Department.

## 2. TABLE OF CONTENTS

Appendices:

DMI only

3. DECK KEY

1. Subroutine MAGT: 1130 Object Deck - sequence # in cc 78-80, 14 cards (BASIC)

2. Test program for MAGT (with control cards and five data cards): 1130 Object Deck - sequence # in cc 78-80, 25 cards (OPTIONAL)

3. Subroutine ILS04: 1130 Object Deck - sequence # in cc 78- 80, 4 cards (BASIC)

4. SUBROUTINE MAGTZ: 1130 Object Deck - sequence # in cc 78-80, 11 cards (BASIC)

5. Test program for MAGTZ (with control cards and five data cards): 1130 Object Deck-sequence # in cc 78 - 80, 22 cards (OPTIONAL)

6. Subroutine IOU: 1130 Object Deck - sequence # in cc 78-80, 3 cards (BASIC)

7. Subroutine REWNZ: 1130 Object Deck - sequence # in cc 78-80, 3 cards (BASIC)

8. Subroutine SFIO: 1130 Object Deck - sequence # in cc 78-80, 24 cards (BASIC)

9. Patch program for Ver. 1, Mod. 4 Fortran Compiler - sequence # in cc 78-80, 5 cards (BASIC)

10. Subroutine MAGTA: 1130 Object Deck - sequence # in cc 78-80, 9 cards (BASIC)

11. Test program for MAGTA (with control cards and five data cards): 1130 Object Deck - sequence # in cc 78-80, 19 cards (OPTIONAL)

12. Complete System Update Deck with Control Cards and Object Decks - 90 cards (OPTIONAL)

## 4. ABSTRACT

This subroutine package includes three main routines - one for use with assembler language programs and two for Fortran compiled programs. The purpose of these routines is to perform standard magnetic tape I/O functions on an 1130 system (running under the 1130 Monitor System) for up to eight series - 2400 magnetic tape units (connected to the CPU via a special RPQ Selector Channel).

The routine for assembler programs conforms to the standard ISS format and conventions used on the 1130 System. Read, Write, Test and associated tape control operations are executed by the routine when it is called by a LIBF sequence in a user's program. The routine utilized standard tape error-checking and recovery procedures and passes error codes to the user's program in the event of errors and/or special conditions (EOT, EOF, etc.). This routine requires the ILS04 ILS subroutine and the MAGT ISS subroutine.

The two routines for use with Fortran programs (but written in assembler language) can be used separately or together in the same user program as desired by the user. Both routines provide read, write, backspace, end file and rewind magnetic tape functions. Error checking and recovery procedures are more limited than in the routine for assembler programs since it was desirable to keep program length to a minimum (however, these procedures can be expanded by the user if it is desirable and if the needed space is available). One routine reads and writes via standard Fortran READ/WRITE statements; hence, all conversion and data formatting provided by the Fortran Compiler is automatically available to the user. The second routine is a called subroutine with the command, tape unit number, data length, and data location as parameters. This routine is quite similar to the first, but moves data directly out of or into core. Hence, it is considerably faster than the first routine, but requires the user to take care of any formatting and conversion that may be necessary for his purposes. These two routines do NOT require the ILS04 routine. However, the first requires the IOU, REWNZ, and the SFIO routines supplied with the package. Also, the first requires that certain recognition sequences in the version 1, Mod. 4 Fortran Compiler be enabled with a "patch" program that is also supplied (on later versions, different compiler changes may be necessary).

This program and its documentation were written by an IBM employee. They have been submitted to the Program Information Department for general distribution in the expectation that they may prove useful to other members of the data processing community. The program and its documentation are, essentially, in the author's original form and have not been subjected to any formal testing. IBM only serves as the distribution agency in supplying this program. It is the user's responsibility to determine the usefulness of and technical accuracy of the program in his own environment. This program is not part of the IBM product line as are Programming Systems (Type I) and Application Programs (Type II).

Questions concerning the use of the program should be directed to the author. Any changes to the program will be reflected in the appropriate Catalog of Programs; however, the changes will not be distributed automatically to users.

CONFIGURATION: (for both assembler and Fortran support)

1130 Monitor System (CPU, disk, card read/punch or paper tape read/punch)

2400 series Magnetic Tape Units (2401's, 2415's, etc.)

2954 RPQ Selector Channel

8K Core

Assembler and/or Fortran Software

DMI
only

## 5-1. SUBROUTINE FOR ASSEMBLER LANGUAGE PROGRAMS (MAGT)

The MAGT subroutine performs all read, write, and control functions relative to IBM 2400 series magnetic tape units. See Figure 5-1. for calling sequence set-up.

### 5-11. Control Parameter

This parameter consists of four hexadecimal digits. See Figure 5-2.

#### I/O Function

The I/O Function digit specifies a particular operation performed on the magnetic tape unit. The functions, associated digital values, and required parameters are listed in Figure 5-3.

#### Test

Branches to LIBF+2 if the previous operation has not been completed, or to LIBF+3 if the previous operation has been completed.

#### Read

Reads the requested number of words into the I/O area from the record at which the tape is positioned. If a read check occurs, the subroutine retries the operation up to 50 times. Each attempt includes backspacing the tape one record and then reading the record. A standard error recovery procedure is used, including checking for noise records and backspacing three records every third attempt. If at any time the record is read correctly, the subroutine exits as if no error occurred.

If a read check still exists after 50 attempts, the subroutine exits to the user's error routine with an error code in the accumulator. Also, if the requested number of words is not equal to the record size, or if a tape mark is read, the subroutine also exits to the user's error routine with an error code in the accumulator. NOTE: The number of words read will never exceed the specified word count.

#### Write With Error Retries

Writes the requested number of words from the I/O area as one record on the specified tape. When the operation is completed, the subroutine determines whether a write check or end-of-tape indicator was encountered. If not, the subroutine exits normally.

If a write check is detected, a retry counter is set for three attempts to write correctly. Each attempt consists of backspacing the tape one record, erasing several inches of tape, and then rewriting that record. If at any time the record is written correctly, the subroutine exits as if no error occurred. If the write check remains after three retries or an end-of-tape indicator is encountered, the subroutine exits to the user's error routine.

#### Write Without Error Retries

Writes the requested number of words from the I/O area as one record on the specified tape. When the operation is completed, the subroutine determines whether a write check or an end-of-tape indicator was encountered. If not, the subroutine exits normally.

If a write check or an end-of-tape indicator was encountered, the subroutine exits to the user's error routine; no rewrites are attempted.

#### Rewind

Initiates a tape rewind and returns control to the user.

#### Rewind and Unload

Initiates a tape rewind and unload and returns control to the user.

#### Backspace

Backspaces one record. If the tape is at the load point marker, no backspace occurs. Note that a backspace does not check for a tape mark.

#### Write Tape Mark

Writes a tape mark on the tape. When the operation is complete, the subroutine processes write checks and end-tape indicators in the same manner as the write with error retries function.

#### Mode Set

The mode set function must be used to change the current status of the control unit and tape drive. This is the only function that uses digits 2 and 3 of the Control Parameter; these digits are ignored for all other functions Refer to SRL Form A22-6866 under mode set commands for a description of setting and resetting mode. Care is urged in using this instruction, since different model tape units have different mode capabilities: incorrect mode commands result in no-ops with NO error indication. Digits 2 and 3 are set according to Figure 5-4.

Device Identification:

This digit specifies which magnetic tape unit is to be used. The digit will be 0-7 corresponding to tape drive zero through seven.

5-12. I/O Area Parameter

The I/O area parameter is the label of the control word which precedes the user's I/O area. This control word contains the word count, which is the number of 16-bit words to be transferred and must not be less than six for a read operation nor less than eight for a write operation.

5-13. Error Parameter

The error parameter is the label of the entry point of the user's error routine. If an error occurs, the subroutine will use a BSI instruction to enter this routine (hence, this label should reference the word just preceding the first instruction of the user's error routine). The user's routine must always return to the tape subroutine via the BSI link. The user should consult SRL Form C26-5929 (IBM 1130 Subroutine Library) before writing this routine to ensure that the requisite conventions are followed under "user's error routine implications". Error handling includes the error branches and recovery choices specified in Appendix A and B. If an error branch occurs for the write or write tape mark functions, the record in error will have been erased; otherwise the tape will be positioned beyond the record in question. A description of terms follows:

Error - Specifies any of the following errors remaining after three retries (write or write tape mark), after fifty retries (read), or after no retries (write without retries): tape data error, program check, or overrun.

EOF - Specifies a tape mark (end-of-file record) read.

EOT - Specifies a tape indicator (end-of-tape reflective marker) sensed during a write or write-tape-mark operation or a tape mark encountered on each of two consecutive read operations.

Long Record - Specifies a partial tape record read since it contained more words than the user's word count.

Short Record - Specifies a tape record read containing fewer words than the user's word count.

Termination - Specifies clearing the routine busy indicator, decrementing the ISS counter (location 50) by 1, and returning to the ILS.

Retry - Specifies initiating another three or fifty retries, according to the function.

Reinitiate - Specifies initiating a read on the next record.

RWU - Specifies initiating a rewind/unload.

Correct Count - Specifies setting the word count in the I/O area to the number actually read.

EOF (under "subroutine action" in Appendix B) - Specifies initiating the writing of one tape mark.

Detailed error procedures are contained in Appendices A and B.

5-14. Sample Program

The MAGT test program reads the first 72 columns from each of five data cards, writes these records on tape unit 0, writes two tape marks, and then rewinds the tape. The records are transferred from unit 0 to unit 1; an extra read is performed on unit 0 so that the first tape mark will be sensed. The reinitiate recovery choice is made, causing the second tape mark to be sensed (thus satisfying the EOT condition) and the RWU/terminate choice is executed. Two tape marks are then written on unit 1 and the tape is rewound, after which the records are read and printed. Five backspace commands are executed, and the records are read and printed a second time. An extra read is performed on unit 1 so that the first of the two tape marks is sensed. The reinitiate choice is executed, causing the second tape mark to be sensed; the RWU/terminate choice is again executed. Tape unit 0 is now spaced forward five records (the operator must reload the tape in response to the 4000 code) by reading five records and an extra read is executed, causing the first tape mark to be sensed; the reinitiate choice is again made, but when the second tape mark is sensed (EOT condition) the terminate choice is made. The fifth record is written on the tape (e.g. beyond the two tape marks), and the tape is backspaced three records. The sequence of reads is again executed, but on EOT, the reinitiate choice is made, causing the block written beyond the tape marks to be read. The tape is then rewound. Another read/print loop is now initiated, during which the RWU/reinitiate choice is executed: the five records are read and printed, the RWU/reinitiate choice is made (after EOT detected), the five records are rea again and printed (the operator must reload unit 0 in response to the 4000 code) and the RWU/terminate choice is made (after EOT detected for the second time). Since the test program is in a read/print loop, the last record is printed a secon time after the RWU/terminate choice.

Finally, the Long and Short Record procedures are tested. A read is executed (the operator must reload unit 0 again) that requests a block shorter than the one on the tape; first, the operation is retried, then it is terminated. The short input block is then printed. Next, a block longer than that on the tape is requested; the correct count/terminate choice is executed and the input block is printed. Finally, the last three blocks are read and printed using the corrected word count, tape 0 is rewound-unloaded, and the program exits.

If at any time a non-correctable read error occurs, the program pauses with/ DEAD in the accumulator: the program should be cancelled and retried in this case. However, if Program Start is pressed, the operation will be retried. The error routines in this test program do NOT check for all possible errors that might occur: if an unexpected error occurs, the test program may hang up in a loop (e.g. a retry loop, etc.). The program should be cancelled and retried in this case.

## 5-15. CONFIGURATION

1130 Monitor System (CPU, disk, card read/punch or paper tape read/punch)

2954 RPQ Selector Channel

2400 Series Tape Units (2401's, 2415's, etc.)

8K Core

## 5-16. SUPPORT

MAGT and ILS04 subroutines only.

Calling Sequence

```
      LIBF      MAGT
      DC        /XXXX      (Control parameter)
      DC        AREA       (I/O Area)
      DC        ERROR      (Error Routine)
      .         .
      .         .
      .         .
```

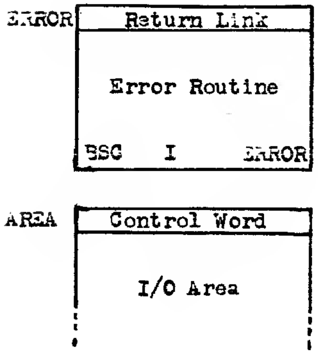


Fig. 5-1.

Control Parameter

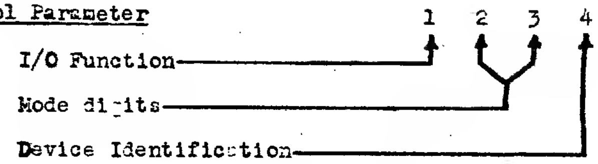


Fig. 5-2.

| Function | Digital Value | Required Parameters* |
|---|---|---|
| Test | 0 | Control |
| Read | 1 | Control, I/O area, Error |
| Write/with error retries | 2 | Control, I/O area, Error |
| Write/without error retries | 3 | Control, I/O area, Error |
| Rewind | 4 | Control |
| Rewind and Unload | 5 | Control |
| Backspace | 6 | Control |
| Write Tape Mark | 7 | Control, Error |
| Mode Set | 8 | Control |

*Any parameters not required for a particular function must be omitted.

Fig. 5-3.

7-track mode digit specifications

| Density(bpi) | Parity | Convert Feature | Translate | Digits 2 | Digits 3 |
|---|---|---|---|---|---|
| 200 | odd | on | off | 1 | 0 |
| 200 | odd | off | off | 3 | 0 |
| 200 | odd | off | on | 3 | 8 |
| 200 | even | off | off | 2 | 0 |
| 200 | even | off | on | 2 | 8 |
| 556 | odd | on | off | 5 | 0 |
| 556 | odd | off | off | 7 | 0 |
| 556 | odd | off | on | 7 | 8 |
| 556 | even | off | off | 6 | 0 |
| 556 | even | off | on | 6 | 8 |
| 800 | odd | on | off | 9 | 0 |
| 800 | odd | off | off | B | 0 |
| 800 | odd | off | on | B | 8 |
| 800 | even | off | off | A | 0 |
| 800 | even | off | on | A | 8 |

9-track mode digit specifications

| Density(bpi) | Digits |
|---|---|
| 800 | C 8 |
| 1600 | C 0 |

Fig. 5-4.

## 5-2. SUBROUTINE FOR FORTRAN COMPILED PROGRAMS (MAGTZ)

The MAGTZ subroutine (when used with the required associated routines and compiler changes as described in 6-132), performs read and write operations with standard Fortran Read/Write statements of the form:

READ (5,n) LIST

Where 5 denotes "ANY magnetic tape, n specifies the format statement, and LIST is a list of variable names. Since standard Read/Write statements are used, all conventional Fortran formatting and data conversion can be used. In addition, backspacing, rewinding, and writing tape marks can be accomplished by use of the statements BACKSPACE n, END FILE n, and REWIND n, where n specifies the desired tape unit. ('Magnetic Tape' must be included in the IOCS card of any Fortran job in which anyof the above tape functions are to be performed.)

## 5-21. WRITE

DMI only

Execution of a Fortran WRITE statement results in a block of 120 characters in packed format being written from the I/O buffer at location/3D onto the tape for each call from the SFIO I/O subroutine (the buffer is in unpacked format, but prior to transfer, each data block is packed). If an error occurs during the operation, a retry counter is set for three attempts to write correctly. Each attempt consists of backspacing the tape one record (i.e. to the beginning of the record in error), erasing several inches of tape, and then rewriting that record. If at any time the record is written correctly, program execution continues as if no error occurred. If the write check remains after three retries, the subroutine pauses with an error code in the accumulator (see Appendix C and 6.2 for error procedures). If the end-of-tape (EOT) reflective marker is sensed during a write operation, two tape marks are written (to signify EOT when the tape is read at a later time) and the tape is rewound-unloaded (see 6.2).

### READ

DMI only

Execution of a Fortran READ statement results in a block of 120 characters being read from the tape and placed into the I/O buffer at location/3D in unpacked format for each call from the SFIO I/O subroutine (each input block is in packed format, but after transfer, each data block is unpacked). If an error occurs during the operation, a retry counter is set for fifty attempts to read correctly. Each attempt consists of backspacing the tape one record (i.e. to the beginning of the record in error) and re-reading that record (any noise records are ignored). If at any time the record is read correctly, program execution continues as if no error occurred. If the read check remains after fifty retries, the subroutine pauses with an error code in the accumulator (see 6.2 and Appendix C for error procedures).
If a tape mark indicating end-of-file (EOF) is sensed during a read operation, the subroutine pauses with EOFX in the accumulator, where X is the number of the tape unit (see 6.2). If tape marks are sensed on two consecutive read operations, the EOT condition is satisfied and the tape is rewound-

unloaded (see 6.2). Hence, the user should always write two tape marks at the end of the last file of data on every tape.

BACKSPACE

Execution of the BACKSPACE n command causes tape unit n to be backspaced one record (if the tape is already at load point, no backspace occurs).

END FILE

Execution of the END FILE n command causes one tape mark to be written on unit n. Error procedures are the same as for WRITE.

REWIND

Execution of the REWIND n command causes tape unit n to be rewound to its load point (if the tape is already at load point, no action is taken).

## 5-22. TAPE UNIT SELECTION

The RPQ Selector Channel for the 1130 can handle up to eight tape units, but only "Magnetic tape' and NOT the specific tape unit desired can be specified in a Fortran READ/WRITE statement; hence, a method of selecting the desired tape unit has been provided. The MAGTZ subroutine maintains a tape unit indicator which is reset each time a BACKSPACE, END FILE, or REWIND command is executed. All read/write operations use this indicator to select the tape unit for that operation.

For example:

```
8     BACKSPACE 1
      READ (5, n) LISTA
      READ (5,m) LISTB
      BACKSPACE 2
      WRITE (5, n) LISTA
      WRITE (5, m) LISTB
      GO TO 8
```

would cause unit 1 to be backspaced one record (no effect if at load point) and LISTA and LISTB to be read from it; then unit 2 would be backspaced one record (again, no effect if at load point) and LISTA and LISTB would be written on it. Now if the operation (i.e. read from unit 1, write on unit 2 ) were to be repeated, a serious inefficiency would result. Unit 1 is now positioned past LISTB; hence, a BACKSPACE 1 would re-position the tape at the beginning of LISTB, so the READ/LISTA command would result in LISTB being read again (to avoid this, an extra read would be necessary). Similarly, the command sequence would cause LISTB on unit 2 to be overwritten with the next record from unit 1. To eliminate this problem, a no-op instruction that resets the unit indicator but causes no tape motion has been provided. When BACKSPACE n, END FILE n, or REWIND n, where n=8 through 15, is encountered, the command is no-oped, but the unit indicator is reset as follows:

| n | unit indicator |
|---|---|
| 8 | 0 |
| 9 | 1 |
| 10 | 2 |
| . | . |
| . | . |
| . | . |
| 15 | 7 |

Hence, the previous example when rewritten becomes:

```
8     BACKSPACE 9
      READ(5,n) LISTA
      READ (5,m) LISTB
      REWIND 10
      WRITE (5,n) LISTA
      WRITE (5,m) LISTB
      GO TO 8
```

ERROR PROCEDURES (EXTENSION)

Error Procedures have been held to a minimum; however, expanded procedures are possible if the user desires (see 7-11).

## 5-23. SAMPLE PROGRAM

The sample program for the MAGTZ subroutine reads the first 72 columns of each of five data cards and writes these records onto tape unit 0. Two tape marks are then written on unit 0 and the tape is rewound. Next, the records are transferred to tape unit 1. An extra read on unit 0 is executed so that the first of the two tape marks will be sensed: the routine pauses with EOFO in the accumulator. The operator should press program start at this time - the routine will execute another read on the next record, which turns out to be another tape mark. Since two consecutive tape marks have been sensed, unit 0 is rewound/unloaded. Two tape marks are now written on unit 2 and this unit is rewound. Finally, the records on unit 2 are read back and written on the printer. An extra read on unit 2 is executed so that the first of the two tape marks will be sensed: the routine pauses with EOFI in the accumulator. The operator should press program start again at this time -- EOT processing will continue as above. The routine then exits via a CALL EXIT. (cf. listing and sample output for MAGTZ test program).

5-24. CONFIGURATION

1130 Monitor System (CPU, Disk, Card Read/Punch or Paper Tape Read/ Punch)

2954 RPQ Selector Channel

Series 2400 Magnetic Tape Units (2401's, 2415's, etc)

8K Core

5-25. SUPPORT

MAGTZ, IOU, REWNZ, SFIO, Fortran Compiler Patch

5-3. SUBROUTINE FOR FORTRAN COMPILED PROGRAMS (MAGTA)

5-31. The MAGTA subroutine is an assembler language routine that can be called from Fortran compiled programs to perform read, write, backspace, end file, and rewind magnetic tape functions. The call instruction for reading and writing is:

CALL MAGTA (n, m, len, name)

where n specifies the command (0=read, 2=write), m specifies the specific tape unit (0-7), 'len' specifies the word count of the data to be transfered, and 'name' is a single variable name specifying the location of the data (the routine transfers 'len' words of data sequentially, starting at location 'name'). The call for backspace, end file, and rewind is:

CALL MAGTA (n, m)

where n and m are as described in the above paragraph. (n=4 backspace; n=5, end file; n=3, rewind).

The advantages of this routine with respect to the MAGTZ routine are: the ability to specify the tape unit directly (rather than with a no-op instruction), a higher rate of data transfer, and the ability to write variable length data blocks (MAGTZ transfers data via the standard Fortran I/O buffer in blocks of 120 characters and interfaces with the SFIO Fortran I/O routine in order to provide formatting and conversion facilities. This sometimes leads to inefficiencies. For example, to transfer an array of 100 integers, the SFIO routine passes only one element at a time into the buffer. Consequently, 100 blocks of 120 characters each are written on tape for the array. The MAGTA routine, on the other hand, transfers the entire array together as a single block of 100 words.)

The major disadvantage of the MAGTA routine is the loss of the formatting and conversion facilities provided by the Fortran compiler via READ/WRITE statements. The MAGTA routine transfers data from core to tape sequentially in core image format: the user must be responsible for formatting and block length.

Both MAGTA and MAGTZ can be used in the same Fortran program; either can be used alone (if MAGTA is used alone, 'MAGNETIC TAPE' should NOT be added to the IOCS cards).

Error procedures for all of the following commands are exactly the same as for the MAGTZ routine (see Appendix C).

WRITE

n=2 'len' words of data are transferred from core to tape unit m sequentially and unchanged, starting at core location 'name'.

READ

n=0 'len' words of data are transferred from tape unit m to core sequentially and unchanged, starting at core location 'name'.

BACKSPACE

n=4 tape unit m is backspaced one record (if at load point, no backspace occurs

END FILE

n=5 a tape mark is written on tape unit m.

REWIND

n=3 tape unit m is rewound to its load point (if at load point, no action is taken)

ERROR PROCEDURES (EXTENSION)

Error procedures have been held to a minimum; however, expanded procedures are possible if the user desires (see 7-11).

5-32. SAMPLE PROGRAM

The sample program for the MAGTA subroutine reads the first 72 columns of each of five data cards and writes these records onto tape unit 0. Two tape marks are then written on unit 0 and the tape is rewound. Next, the records are transferred to tape unit 1. An extra read on unit 0 is executed so that the first of the two tape marks will be sensed: the routine pauses with EOFO in the accumulator. The operator should press program start at this time -- the routine will execute another read on the next record, which turns out to be another tape mark. Since two consecutive tape marks have been sensed, unit 0 is rewound/unloaded. Two tape marks are now written on unit 2 and this unit is rewound. Finally, the records on unit 2 are read back and written on the printer. An extra read on unit 2 is executed so that the first of the two tape marks will be sensed: the routine pauses with EOFI in the accumulator. The operator should press program start again at this time -- EOT processing will continue as above. The routine then exits via a CALL EXIT. (cf. listing and sample output for MAGTA test program).

## 33. CONFIGURATION

1130 Monitor System (CPU, disk, card read/punch or paper tape read/punch)

2954 RPQ Selector Channel

Series 2400 Magnetic Tape Units (2401's, 2415's, etc.)

8K Core

## 34. SUPPORT

MAGTA

## 6-1. SYSTEM SET-UP

### 6-11. HARDWARE

1130 Monitor System (CPU, disk, card read/punch or paper tape read/punch), 2400 series tape units (2401's, 2415's, etc.), 2954 RPQ Selector Channel, 8K core.

NOTE: The Tape Control Unit address should be set to 8. The tape units should have addresses 0-7.

### 6-12. SOFTWARE

Assembler and/or Fortran software

### 6-13. SUPPORT

6-131. MAGT System -

| | |
|---|---|
| Subroutines required: | MAGT<br>ILS04 |

Procedure: the 1130 subroutine library must have the MAGT and ILS04 routines added to it. One update deck only is required (see Figure 6-1). If only object decks are supplied, just add the indicated control cards. Updating job is run just as any ordinary job, either stacked with other jobs or alone with a cold start card.

6-132. MAGTZ System -

| | |
|---|---|
| Subroutines required: | MAGTZ<br>IOU<br>REWNZ<br>SFIO<br>Fortran Compiler Patch |

Procedure: the 1130 subroutine library must have the MAGTZ, IOU, REWNZ, and SFIO routines added to it; in addition the Fortran compiler must be patched (the version 1, mod. 4 compiler requires only that certain recognition sequences be enabled -- newer versions may require different patching from that which is presented here). The updating and patching job is run just as any ordinary job, either stacked with other jobs or alone with a cold start card (see Figure 6-2.). If only the object decks are supplied, just add the indicated control cards.

6-133. MAGTA System -

Subroutines required: MAGTA

Procedure: the 1130 subroutine library must have the MAGTA routine added to it. One update deck only is required (see Figure 6-3.). If only the object deck is supplied, just add the indicated control cards. Updating job is run just as any ordinary ;ob, either stacked with other jobs or alone with a cold start card.

## 6-2. ERROR HALTS AND PROCEDURES

Error conditions, codes, and user/operator procedures are detailed in Appendixes A, B, and C.

## 6-3. TAPE UNIT OPERATION

Reloading a tape always causes a level 4 interrupt; hence, care must be taken to avoid reloading a tape at a time when the proper routines for handling the interrupt are NOT in core (e.g. while the system is being loaded, while a new job is being loaded or compiled, between stacked jobs, etc.). An easy method to do this is to always wait to reload the required tapes until the program displays the tape "not ready" code in the accumulator. Users unfamiliar with magnetic tape device operations should read 'IBM System/360 Component Description 2400 - Series Magnetic Tape Units and 2816 Switching Unit' (A22-6866-3) Page 4-11, (Magnetic Tape Unit Principles), and Page 34-48 (2400 Tape Unit Keys and Lights; Tape Handling and Organization, Tape Unit Loading and Unloading Procedures).

Except for the above procedures (6-2. and 6-3.), no special console settings, etc. are required.

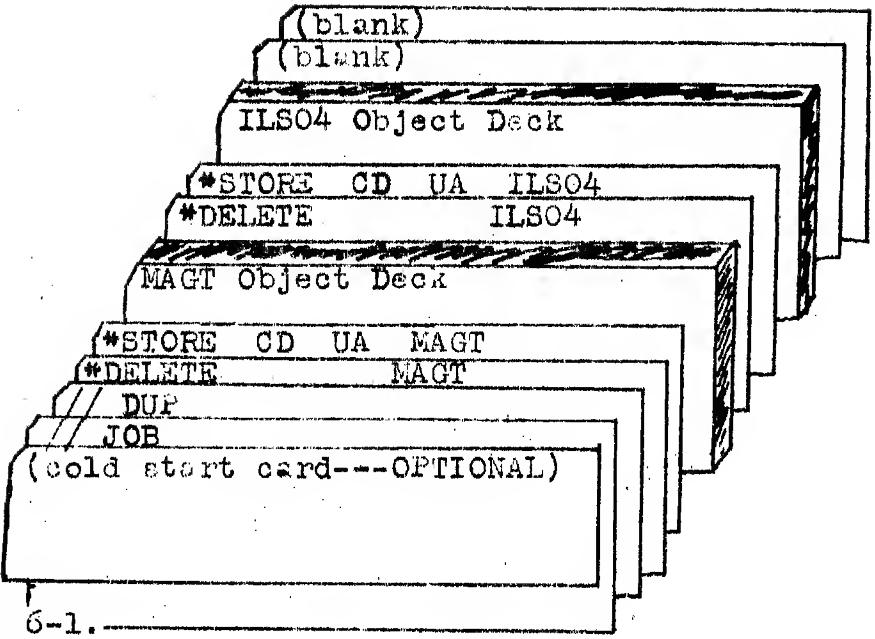

(blank)
(blank)
ILSO4 Object Deck
*STORE CD UA ILSO4
*DELETE ILSO4
MAGT Object Deck
*STORE CD UA MAGT
*DELETE MAGT
// DUP
// JOB
(cold start card---OPTIONAL)

Fig. 6-1.

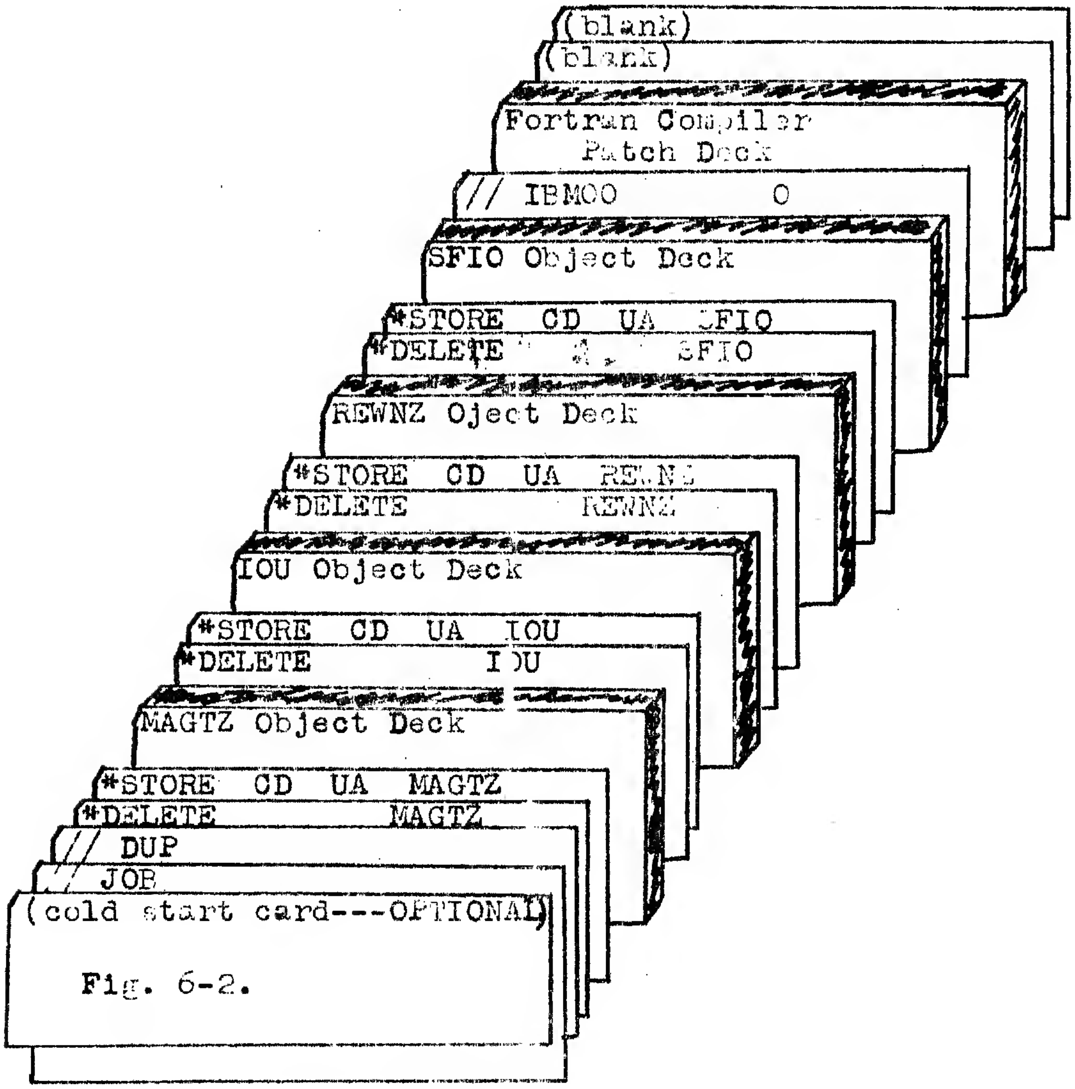

(blank)
(blank)
Fortran Compiler Patch Deck
// IBMOO 0
SFIO Object Deck
*STORE CD UA SFIO
*DELETE SFIO
REWNZ Oject Deck
*STORE CD UA REWNZ
*DELETE REWNZ
IOU Object Deck
*STORE CD UA IOU
*DELETE IOU
MAGTZ Object Deck
*STORE CD UA MAGTZ
*DELETE MAGTZ
// DUP
// JOB
(cold start card---OPTIONAL)

Fig. 6-2.

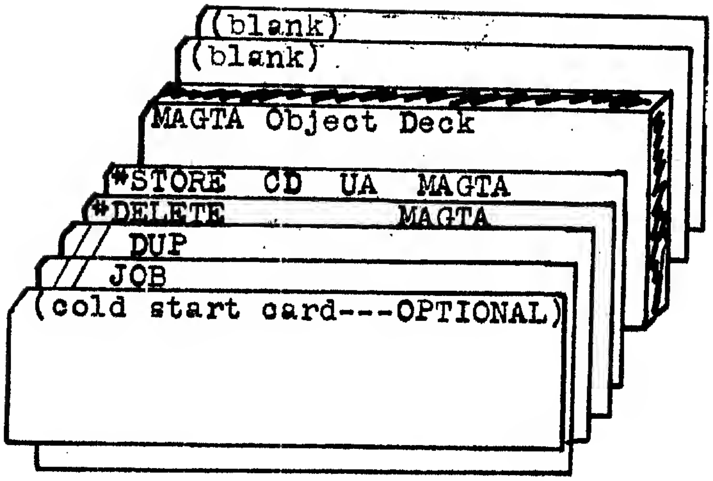

(blank)
(blank)
MAGTA Object Deck
*STORE CD UA MAGTA
*DELETE MAGTA
// DUP
// JOB
(cold start card---OPTIONAL)

Fig. 6-3.

## 7-1. POSSIBLE PROGRAM MODIFICATIONS

### 7-11. EXPANDED ERROR PROCEDURES (MAGTZ, MAGTA)

1. cf. label "A" - in a similar manner to the present coding, the user can set-up DEDX (to be stored in 'FBADA') at this point, instead of having just "DEAD".

2. cf. label 'B' - insert:

```
BSI          TREDY
LD           DATA
SLA          14
BSC   L      PRO, -
LIBF         PAUSE
DC           FPRCT

PRO (next instruction)
```

and add set-up for FEFX (to be stored in 'FPRCT') at 'A'. The above coding will display FEFX if a tape is file-protected on a write command. The user can terminate the job, or can replace the file - protect ring and press program start, which will cause the write command to be executed. (Note: the above coding may necessitate some addressing changes in other sections of the program.)

3. cf label 'WTEOR' - change the coding as follows:

```
WTEOR      LIBF                PAUSE
           DC                  FEOTD
BRN        MDX                 *
           MDX         L       C003,-1
           MDX                 TMEOT
           MDX         L       C003, +3
           MDX                 RWU
```

and add set-up for FEOX (to be stored in 'FEOTD') at 'A'.

The above coding will display FEOX when the end-of-tape marker is sensed during the execution of a write or write tape mark command. If the user presses program start, normal EOT action will be taken; if the user puts /70FB into 'BRN' from the console and then presses program start, the routine will exit without writing the tape marks or unloading the tape (hence, blocks could be written beyond the EOT marker). If another write or write tape mark command is executed (but before a backspace, which would reset the EOT indicator), the routine will again pause with FEOX in the accumulator. If the user now wants to execute normal EOT procedures, he must put /7000 into 'BRN' and press program start.

4. cf label 'A' and 'PERM' - for the non-correctable read/write error message, the user could set-up /BDNX (to be stored in 'FBAD') at 'A' so that N denoted read, write, or write tape mark and X denoted the tape unit. In addition, the coding at 'PERM' could be changed in a manner similar to the change noted in 3. above, so that the operator could cause a branch to 'ERROR', thus causing the operation to be retried when program start is pressed.

NOTE: the user could write his own LIBF routines to act as error routines: the LIBF calls would replace the LIBF PAUSE calls. Then these error routines could do the necessary program resetting without the need for operator intervention.

### 7-12. WORD COUNT TO BYTE COUNT CONVERSION (MAGT)

For some applications, it may be desirable for the user to be able to specify a byte count rather than a word count. The 2954 RPQ Selector Channel transfers data on an even byte count. If the count is odd and the command is write, the rightmost byte of the last word is ignored and just the desired number of bytes is transferred; however, if the command is read, the rightmos byte of the last word is zeroed -- hence, this last byte must be saved and restored when the count is odd and the command is read. The following coding will accomplish this.

delete the SLA 1 command from location labelled 'ONE'
delete the SRA 1 command from location labelled 'TWO'

```
just before 'MTBEN', insert     LD             INITA
                                BSC    L       ODD,E

just before 'BYTCT', insert     LD             INITA
                                BSC    L       ODSET,E
```

at the end of the program, insert:

```
ODD        SRA            1
           A              INITA+2
           STO            LOAD+1
LOAD       LD       L     *-*
           AND            OOFF (label 'MTMK3')
           STO      L     LASTW
           BSC      L     MTBEN
ODSET      LD       I     LOAD+1
           OR             LASTW
           STO      I     LOAD+1
           BSC      L     BYTCT
LASTW      DC             0
```

at location labelled 'THREE', replace S MT006 with S MTCMN.

```
// JOB
// ASM
*LIST   7-21.
*PRINT SYMBOL TABLE
*LEVEL 4

                                  LIBR
0000       140478C0               ISS   05 MAGT          4
0000 0   6A17             MAGT    STX    2 MTRET+1         LIBF ENTRANCE
0001 00  66800000                 LDX   I2 0               LOAD A(LIB+1)
0003 0   7004                     MDX       *+4
0004 0   0000             MINT    DC        0              INTERRUPT ENTR
0005 01  4C0000D7                 BSC   L   MTRRR
0007 0   0001                     DC        1
0008 0   6911                     STX    1 MTRET+3         SAVE XR1
0009 01  6500009A                 LDX   L1 MTSV            SET ADDRESSING
000B 0   D900                     STD    1 0               SAVE ACC & EXT
000C 0   280E                     STS       MTRET+4        SAVE STATUS
000D 0   C200                     LD     2 0               LOAD CONTROL P
000E 0   180C                     SRA       12             ISOLATE FUNCT.
000F 01  740000A1                 MDX   L   MTBSY,0        TEST ROUTINE B
0011 0   700C                     MDX       MTRET+7        NOT BUSY,BRANC
0012 01  4C20000F                 BSC   L   *-5,Z          BUSY, LOOP IF
0014 0   7201                     MDX    2 +1              FORM LIBF+2
0015 0   6A07                     STX    2 MTRET+6         FSTORE RETURN
0016 0   C900                     LDD    1 0               RESTORE ACC &
0017 00  66000000         MTRET   LDX   L2 0               RESTORE XR2
0019 00  65000000                 LDX   L1 0               RESTORE XR1
001B 0   2000                     LDS       0              RESTORE STATUS
001C 00  4C400000                 BOSC  L   0              EXOT TO USER/I
001E 01  4C200022                 BSC   L   *+2,Z          IF NOT TEST, C
0020 0   7201                     MDX    2 +1              FORM LIBF+2
0021 0   70F2                     MDX       MTRET-3        RETURN VIA LIB
0022 0   6A7A                     STX    2 MTSV+3          STORE A(LIBF+1
0023 01  74FF00A1                 MDX   L   MTBSY,-1       SET ROUTINE BU
0025 0   1000                     NOP
0026 0   D13A                     STO    1 MTFUN-MTSV      SAVE FUNCTION
0027 0   D13B                     STO    1 RWRSW-MTSV      SET READ/WRITE
0028 0   910A                     S      1 MTFMX-MTSV      TEST FUNCTION
0029 01  4C300063                 BSC   L   MTILL,Z-       IF+, ILLEGAL F
002B 0   8159                     A      1 MTRGO+1-MTSV    RESTORE FUNC.
002C 0   D01E                     STO       MTGO           STORE FUNCT(MD
002D 0   D158                     STO    1 MTRGO-MTSV      SET RECOV ENTR
002E 0   C200                     LD     2 0               RELOAD CONTROL
002F 0   E10D                     AND    1 MTOUF-MTSV      ISOLATE DEVICE
0030 0   E90C                     OR     1 MTMK7-MTSV      FORM DD8X
0031 0   D119                     STO    1 INIT+1-MTSV     STORE IN IOCC
0032 0   D121                     STO    1 TSSEA+1-MTSV    STORE IN XSENS
0033 0   D10F                     STO    1 GEST+1-MTSV     STORE IN RECOV
0034 0   100C                     SLA       12
0035 0   1804                     SRA       4
0036 0   D13B                     STO    1 MTINT-MTSV      /0X00
0037 0   C108                     LD     1 TSRET-MTSV
0038 0   D157                     STO    1 ILSGO+1-MTSV    SET RETURN
0039 0   0920             MTCSS   XIO    1 TSSEA-MTSV      FETCH SENSE DA
003A 0   70FF                     MDX       *-1            WAIT INTERUPT
003B 0   C116             WARET   LD     1 MTUST-MTSV      LOAD UNIT STAT
003C 0   E136                     AND    1 M0050-MTSV      ISOLATE BUSY,          BITS
003D 01  4C600039                 BOSC  L   MTCSS,Z        IF BOTH OFF, C
003F 0   C13C                     LD     1 CSTAT-MTSV      LOAD CHANL STA
0040 01  4C280063                 BSC   L   MTILL,+Z       IF NON-EXIST,
0042 0   C11D                     LD     1 TSDAT-MTSV      LOAD SENSE DAT
0043 0   100A                     SLA       10             SET TU-A, TU-B
0044 01  4C020047                 BSC   L   REDY,C         IF READY, BRAN
0046 0   7C19                     MDX       MTNR           NOT READY, EXI
```

```
0047 01 4C680039     REDY  BOSC L  MTCSS,Z+        IF BUSY, RETES
0049 01 6680009D           LDX  I2 MTSV+3          RESTORE A(LIBF
004B 0  7000         MTGO  MDX     *               INITIAL BRANCH
004C 0  7021               MDX     MTRD            READ
004D 0  7025               MDX     MTWEN           WRITE/W
004E 0  7024               MDX     MTWEN           WRITE/WOUT
004F 0  7009               MDX     MTLP            REWIND
0050 0  7038               MDX     MTIEN           REWIND-UNLOAD
0051 0  7007               MDX     MTLP            BSP
0052 0  7033               MDX     MTBEN           WRITE TAPE MAR
0053 00 C6800000           LD   I2 0               LOAD CONTROL P
0055 0  1804               SRA     4               POSITION CODE
0056 0  E069               AND     MTMK3           FORM 000X+3
0057 0  E937               OR    1 MT003-MTSV
0058 0  7034               MDX     MTIEN+4        PROCEED TO STOR
0059 0  C05D         MTLP  LD      TSDAT           IF AT LOAD PT,
005A 0  100C               SLA     12                    BACKSPACE        D
005B 01 4C100089           BSC  L  MTIEN,-
005D 01 740100A1           MDX  L  MTBSY,+1
005F 0  70B4               MDX     MTRET-3
0060 0  C13B         MTNR  LD    1 MTINT-MTSV      LOAD 0X00
0061 0  E85F               OR      MTMK6           FORM 4X00
0062 0  7001               MDX     MTILL+1         EXIT THRU DV N
0063 0  C062         MTILL LD      MTECD           LOAD ILLEGAL C
0064 01 6680009D           LDX  I2 MTSV+3          RELOAD A(LIBF+
0066 0  72FF               MDX   2 -1              FORM A(LIBF)
0067 00 6E000028           STX  L2 40              STORE A(LIBF)
0069 0  6229               LDX   2 41              SET 41 AS RETU
006A 0  6AB2               STX   2 MTRET+6
006B 01 740100A1           MDX  L  MTBSY,+1        SET ROUTINE NT
006D 0  70A9               MDX     MTRET
006E 0  6233         MTRD  LDX   2 51              SET RETRY CNT
006F 0  6A3A               STX   2 RECNT
0070 0  6210               LDX   2 16              SET READ MIN
0071 0  6A60               STX   2 RWRSW           SET READ/WRITE
0072 0  7(03               MDX     *+3
0073 0  6204         MTWEN LDX   2 4               SET WRITE CNT
0074 0  6A35               STX   2 RECNT
0075 0  620C               LDX   2 12              SET WRITE MIN
0076 0  6A25               STX   2 MTSV+2          SAVE MIN
0077 01 6680009D           LDX  I2 MTSV+3          RELOAD A(LIBF+
0079 00 C6800001           LD   I2 1               LOAD WORD CNT
007B 0  D024               STO     MTSV+6          SAVE WORD COUN
007C 0  1001      ONE ---> SLA     1               MULT COUNT=BYT
007D 0  D036               STO     INITA           STORE BYTE COU
007E 0  901D               S       MTSV+2          IS CNT OVER MI
007F 01 4C280063           BSC  L  MTILL,Z+        IF NO, BRANCH
0081 0  7201               MDX   2 +1              FORM LIBF+2
0082 0  C200               LD    2 0
0083 0  D01B               STO     MTSV+5          SAVE A(AREA)
0084 0  8027               A       M1              INCRM. TO A(EF
0085 0  D030               STO     INITA+2         STORE A(AREA)
0086 0  C201         MTBEN LD    2 1               LOAD A(ERR)
0087 0  D016               STO     MTSV+4          SAVE A(ERR)
0088 0  7201               MDX   2 +1              FORM LIBF+3
0089 01 658000D4     MTIEN LDX  I1 MTFUN
008B 01 C50000C6           LD   L1 MTCCS-1         SET CODE
008D 0  D027               STO     INITA+1         INTO CCW
008E 0  C01F               LD      AILL2           RESET ILSGO AD
008F 0  D061               STO     ILSGO+1
0090 0  71FF               MDX   1 -1              TEST FOR READ
0091 0  6841               STX     EOTSW           SET EOT SWT IF
```

```
0092 00 74000032          MDX  L  50,0
0094 0  7002              MDX     *+2
0095 00 74010032          MDX  L  50,+1          INCRM ISS COUN
0097 0  081A        EXEC  XIO     INIT           INITIATE I/O O
0098 01 4C000014          BSC  L  MTRET-3        RETURN TO USER
009A    0007        MTSV  BSS  E  7              STORAGE AND CO
00A1 0  0001        MTBSY DC      1
00A2 1  003B        TSRET DC      WARET
00A3 0  DF03        TSCSW DC      /DF03
00A4 0  0008        MTFMX DC      8
00A5 0  0000        MTWSV DC      0
00A6 0  DD80        MTMK7 DC      /DD80
00A7 0  000F        MT00F DC      /000F
00A8 1  00AA        GEST  DC      GEST+2
00A9 0  DD00              DC      /DD00
00AA 0  0000        RECNT DC      0
00AB 0  0000              DC      0
00AC 0  0001        M1    DC      1
00AD 0  DF06        INSTA DC      /DF06
00AE 1  00F2        AILL2 DC      ILSGO+2
00AF 0  DF00        INSTB DC      /DF00
00B0 0  0000        MTUST DC      0
00B1 0  000C        MTCMN DC      12
00B2 1  00B4        INIT  DC      INITA
00B3 0  0000              DC      0
00B4 0  0000        INITA DC      0
00B5 0  0000              DC      0
00B6 0  0000              DC      0
00B7    0003        TSDAT BSS     3
00BA 1  00BC        TSSEA DC      TSSEA+2
00BB 0  0000              DC      0
00BC 0  0006        MT006 DC      6
00BD 0  0004        MT004 DC      4
00BE 1  00B7              DC      TSDAT
00BF 0  0011        MTMK2 DC      /0011
00C0 0  00FF        MTMK3 DC      /00FF
00C1 0  4000        MTMK6 DC      /4000
00C2 0  000A        M10   DC      10
00C3 0  0003        BSPCT DC      3
00C4 0  0004        BSPSW DC      4
00C5 0  0003        FSPSW DC      3
00C6 0  4001        MTECD DC      /4001
00C7 0  0002        MTCCS DC      /0002
00C8 0  2001              DC      /2001
00C9 0  2001              DC      /2001
00CA 0  0007              DC      /0007
00CB 0  000F        RWUC  DC      /000F
00CC 0  0027        BSPC  DC      /0027
00CD 0  001F        TMC   DC      /001F
00CE 0  0017        ERASC DC      /0017
00CF 0  0037        FSPC  DC      /0037
00D0 0  0050        M0050 DC      /0050
00D1 0  0003        MT003 DC      3
00D2 0  0000        RWRSW DC      0
00D3 0  0001        EOTSW DC      1
00D4 0  0000        MTFUN DC      0
00D5 0  0000        MTINT DC      0
00D6 0  0000        CSTAT DC      0
00D7 0  08D6        MTRRR XIO     INSTB-1        TEST CHANL STA
00D8 01 6500009A          LDX  L1 MTSV
00DA 0  6A3A              STX   2 TEMP+1
00DB 0  6200              LDX   2 0              INITIALIZE ERR
```

```
00DC 0   700F         SKP    MDX       OVER
00DD 0   0000         TENSE  DC        0
00DE 01  74F600DC            MDX   L   SKP,-10
00E0 0   0920                XIO    1  TSSEA-MTSV      FETCH SENSE DA
00E1 0   7032                MDX       TEMP
00E2 0   08BF                XIO       TSCSW-1         FETCH UNIT STA
00E3 01  740A00DC            MDX   L   SKP,+10
00E5 0   C11D                LD     1  TSDAT-MTSV
00E6 01  4C9000DD            BSC   I   TENSE,-         IF COM REJ, GO           CODE
00F8 01  660001D4            LDX   L2  RTST+2
00EA 01  4C000192            BSC   L   RWUT+3
00EC 0   D0E9         OVER   STO       CSTAT
00ED 0   08B4                XIO       TSCSW-1         FETCH UNIT STA
00FE 0   D0C1                STO       MTUST
00EF 0   100E                SLA       14              SET UC, UE BIT
00F0 01  640000F2     ILSGO  LDX   L   *               BRANCH TO INT
00F2 0   7000         MTRGO  MDX       *
00F3 0   7007                MDX       READ
00F4 0   7033                MDX       WOWTM
00F5 0   7C25                MDX       WWOR
00F6 0   7(21                MDX       EXITA
00F7 0   7C20                MDX       EXITA
00F8 0   701F                MDX       EXITA
00F9 0   702F                MDX       WOWTM
00FA 0   701D                MDX       EXITA
00FB 0   C0B4         READ   LD        MTUST           CHK FOR TM(EOF OR EOT)
00FC 01  4C04015D            BSC   L   MTEOF,E         UE ON(ODD), BR
00FF 0   D0D4                STO       EOTSW
00FF 0   08AC         BYTCT  XIO       INSTA-1         FETCH BYTE CNT           E
0100 0   80B3                A         INITA           SUBTR CCW COUN
0101 0   90AA                S         M1              ADJUST ACTUAL
0102 0   1801       TWO →    SRA       1
0103 0   D0A1                STO       MTWSV           SAVF CORRECT C
0104 0   90B7       THREE →  S         MT006
0105 01  4C2801D2            BSC   L   RTST,+Z         IF NOISE, REIN
0107 0   C0A8         F      LD        MTUST           RELOAD UNIT ST
0108 0   100E                SLA       14              SET UC BIT
0109 01  4C28015A            BSC   L   M,+Z            IF ON, BRANCH TO RETRY
010B 0   C0CA                LD        CSTAT           FETCHCHANL STA
010C 0   1006                SLA       6               SET LENGTH BIT
010D 01  4C280177            BSC   L   LORSH,+Z        IF ON(NEG),BRA
010F 01  740100A1     EXIT   MDX   L   MTBSY,+1        SET ROUTINE NO
0111 00  74FF0032            MDX   L   50,-1           DECRM ISS COUN
0113 0   1000                NOP
0114 00  66000000     TEMP   LDX   L2  0               RESTORE XR2 AN
0116 01  4CF00004            BSC   I   MINT            RETURN TO USER
0118 01  4C02010F     EXITA  BSC   L   EXIT,C          IF DE ON (ODD), EXIT
011A 0   70F9                MDX       TEMP            IF NO, AWAIT S           R.
011B 01  4C280121     WWOR   BSC   L   ERRA,+Z         IF UC ON, ERR
011D 0   C116         NOER   LD     1  MTUST-MTSV      LOAD UN STAT
011E 01  4C040125            BSC   L   MTWOT,E         IF EOT , BRANC
0120 0   70FE                MDX       EXIT            TERMINATE IF N
0121 0   40BB         ERRA   BSI       TENSE           CHK FOR COM RE
0122 0   620E                LDX    2  14              TERM IF NT EOT
0123 0   4029                BSI       CDSET           INDICATE ERROR
0124 0   70F8                MDX       NOER
0125 0   620F         MTWOT  LDX    2  15              LOAD WWOR EOT
0126 0   4026                BSI       CDSET           USER VIA ACTIO
0127 0   70E7                MDX       EXIT            TERMINATE
0128 01  4C280131     WOWTM  BSC   L   ERRB,+Z         IF UC ON, ERR
012A 0   C116         NOTER  LD     1  MTUST-MTSV      LOAD UN STAT
012B 01  4C04012E            BSC   L   *+1,E           IF EOT, BRANCH
```

```
012D 0  70E1                MDX      EXIT              IF NOT EOT, EX
012E 0  620C                LDX    2 12                SET EOT CODE
012F 0  401D                BSI      CDSET             INFORM USER
0130 0  7018                MDX      FUTRY
0131 0  40AB          ERRB  BSI      TENSE             CHK FOR COM RE
0132 0  7002                MDX      *+2
0133 0  C123          H     LD     1 TSSEA+3-MTSV      SET RETRY COUN
0134 0  D110                STO    1 RECNT-MTSV
0135 0  4079                BSI      RETRY
0136 0  C116                LD     1 MTUST-MTSV        LOAD UN STAT
0137 01 4C04013D            BSC    L EOTON,E           IF EOT, BRANCH
0139 0  620B          ERALO LDX    2 11                SET ERROR CODE
013A 0  4012                BSI      CDSET             INFORM USER
013B 0  4073                BSI      RETRY
013C 0  70F9                MDX      ERALO-3
013D 0  620D          EOTON LDX    2 13                SET ERR/EOT CO
013E 0  400E                BSI      CDSET
013F 01 4C280133            BSC    L H,+Z              RETRY
0141 01 4C040149            BSC    L FUTRY,E           EOF/RWU/TERM
0143 01 440001A4            BSI    L WTM               EOF/RWU/RETRY
0145 01 44000199            BSI    L RWU
0147 0  4047                BSI      RWUT              AWAIT RELOADIN
0148 0  70EA                MDX      H
0149 0  405A          FUTRY BSI      WTM               EOF/RWU/TERM
014A 0  404E                BSI      RWU
014B 0  70C3                MDX      EXIT
014C 0  0000          MTSAV DC       0
014D 0  0000          CDSET DC       0                 RETURN LINK
014E 0  C13B                LD     1 MTINT-MTSV        LOAD 0X00 DEVI
014F 0  6AFC                STX    2 MTSAV             SAVE ERR CODE
0150 0  80FB                A        MTSAV             FORM 0X0M(FULL
0151 01 4480009E            BSI    I MTSV+4            GO TO USERS ER
0153 0  4F18                BSC      +-                USERS RETURN,
0154 0  70BA                MDX      EXIT              IF ZERO, TERM
0155 01 4C80014D            BSC    I CDSET             IF NO, RECOVER
0157 0  6233          RERE  LDX    2 51                RESET RETRY CN
0158 01 6E0000AA            STX   L2 RECNT
015A 0  4054          M     BSI      RETRY
015B 0  40F1          ERR   BSI      CDSET             ERROR ALONE-CH
015C 0  70FA                MDX      RERE              RETRY
015D 01 740000D3      MTEOF MDX    L EOTSW,0           LAST COMM SENS
015F 0  700B                MDX      EOF               IF NO, SET EOF
0160 0  6206          FOFOT LDX    2 6                 SET EOF/EOT CO
0161 0  40EB                BSI      CDSET
0162 01 4C280170            BSC    L RWREI,+Z          RWU/REINIT
0164 01 4C0401D2            BSC    L RTST,E            REINIT
0166 0  4032          RWTM  BSI      RWU               RWU/TERM
0167 0  70A7                MDX      EXIT
0168 01 4C0201D2      BRN   BSC    L RTST,C             DE ON
016A 0  70A9                MDX      TEMP              IF DE NT ON, A              R
016B 0  1010          EOF   SLA      16
016C 0  D139                STO    1 EOTSW-MTSV        SET EOT SWITCH
016D 0  6202                LDX    2 2                 SET EOF ALONE
016E 0  40DE                BSI      CDSET             GO TO USER FOR
016F 0  7062                MDX      RTST              REINITIATE
0170 0  4028          RWREI BSI      RWU               RWDUN/REINIT
0171 0  401D                BSI      RWUT              AWAIT RELOADIN
0172 0  705F                MDX      RTST
0173 0  C108          CWCTM LD     1 MTWSV-MTSV        LAAD ACTUAL CN
0174 01 D480009F            STO    I MTSV+5            STORE IN USER
0176 0  7098                MDX      EXIT              TERMINATE
0177 0  0912          LORSH XIO    1 INSTA-1-MTSV      CHK FOR LENGTH
```

```
0178 01 4C30017D         BSC  L  LONG,-Z          IF +, BRANCH C
017A 0  6208             LDX   2 8                SHORT ALONE
017B 0  40D1             BSI     CDSET            SHORT INPUT RECORD
017C 0  70F6             MDX     CWCTM            CORRECT WRD CN
017D 0  6207     LONG    LDX   2 7                LONG INPUT RECORD
017E 0  40CE             BSI     CDSET
017F 0  70D7             MDX     RERE             RETRY
0180 0  D111     GSTAR   STO   1 GEST+3-MTSV      EXEC BKSP, FSP
0181 0  090E             XIO   1 GEST-MTSV               RWU, O
0182 0  7052             MDX     T
0183 01 6600018B WRT     LDX  L2 WSP              WRITE RETRY
0185 0  6AE3             STX   2 BRN+1
0186 0  C11D             LD    1 TSDAT-MTSV       FETCH SENSE DA
0187 0  1809             SRA     9
0188 01 4C04018B         BSC  L  WSP,E            SKIP BSP IF NO
018A 0  704D             MDX     BSONE            GO TO BKSP
018B 01 74030169 WSP     MDX  L  BRN+1,+3
018D 0  704E             MDX     ERASE
018E 0  7043             MDX     RTST
018F 0  0000     RWUT    DC      0                AFTER RWU/RETR         OICES
0190 01 66000197         LDX  L2 BACK                WAIT AT 41          D
0192 00 6E000028         STX  L2 40
0194 0  C13B             LD    1 MTINT-MTSV
0195 0  E927             OR    1 MTMK6-MTSV
0196 0  6029             LDX     41               AWAIT UNIT REL
0197 01 4C80018F BACK    BSC  I  RWUT
0199 0  0000     RWU     DC      0                RWU ROUTINE
019A 01 660001A2         LDX  L2 RWURE
019C 0  6ACC             STX   2 BRN+1
019D 0  C131             LD    1 RWUC-MTSV
019E 0  D111     GO      STO   1 GEST+3-MTSV
019F 0  C037             LD      ARENT
01A0 0  D157             STO   1 ILSGO+1-MTSV
01A1 0  70DF             MDX     GSTAR+1
01A2 01 4C800199 RWURE   BSC  I  RWU
01A4 0  0000     WTM     DC      0                WTM ROUTINE
01A5 01 660001AA         LDX  L2 WTMRE
01A7 0  6AC1             STX   2 BRN+1
01A8 0  C133             LD    1 TMC-MTSV
01A9 0  70F4             MDX     GO
01AA 01 660001AD WTMRE   LDX  L2 MTW2
01AC 0  70FA             MDX     WTM+1
01AD 01 4C8001A4 MTW2    BSC  I  WTM
01AF 0  0000     RETRY   DC      0                MAIN RETRY ENT
01B0 01 74FF00AA         MDX  L  RECNT,-1         RETRY FINISHED
01B2 0  7003             MDX     *+3              IF NO, RETRY
01B3 0  6201             LDX   2 1                SET ERROR CODF
01B4 01 4C8001AF         BSC  I  RETRY            RETURN
01B6 0  C020             LD      ARENT            RESET ILSGO AD
01B7 0  D157             STO   1 ILSGO+1-MTSV
01B8 0  C138             LD    1 RWRSW-MTSV
01B9 0  100C             SLA     12
01BA 01 4C200183         BSC  L  WRT,Z            IF NOT ZERO, I
01BC 01 660001D2         LDX  L2 RTST             READ RETRY
01BE 0  6AAA             STX   2 BRN+1
01BF 01 74FF00C3 RSP     MDX  L  BSPCT,-1         TEST BSP CNT
01C1 0  7016             MDX     BSONE            IF 1 BSP, BRAN
01C2 01 74F20169         MDX  L  BRN+1,-14        RESET ENTRY
01C4 01 74FF00C4         MDX  L  BSPSW,-1         3 BSP COMPLETE
01C6 0  7011             MDX     BSONE            IF NO, BSP AGA
01C7 01 74050169         MDX  L  BRN+1,+5         IF YES, RESET
01C9 01 74FF00C5         MDX  L  FSPSW,-1         2 FSP COMPLETE
```

```
01CB 0  700E              MDX     FSONE             IF NO, FSP AGA
01CC 01 740300C3          MDX   L BSPCT,+3
01CE 01 740300C5          MDX   L FSPSW,+3
01D0 01 740400C4          MDX   L BSPSW,+4
01D2 0  C114       RTST   LD    1 AILL2-MTSV        EXEC RETRY OR
01D3 0  D157              STO   1 ILSGO+1-MTSV      RESET ILSGO AD
01D4 0  0918              XIO   1 INIT-MTSV
01D5 01 4C000114   T      BSC   L TEMP
01D7 1  0168       ARENT  DC      BRN
01D8 0  C132       BSONE  LD    1 BSPC-MTSV         SET
01D9 0  70A6              MDX     GSTAR               APPROPRIATE
01DA 0  C135       FSONE  LD    1 FSPC-MTSV             COMMAND
01DB 0  70A4              MDX     GSTAR                   FOR
01DC 0  C134       ERASE  LD    1 ERASC-MTSV                GSTAR
01DD 0  70A2              MDX     GSTAR
01DE                      END
```

```
                                  SYMBOL  TABLE

AILL2  00AE     ARENT  01D7     BACK   0197     BRN    0168     BSONE  01D8
BSPC   00CC     BSPCT  00C3     BSPSW  00C4     BYTCT  00FF     CDSET  0140
CSTAT  00D6     CWCTM  0173     E      0107     EOF    016B     EOFOT  0160
EOTON  013D     EOTSW  00D3     ERALO  0139     ERASC  00CE     ERASE  01DC
ERR    015B     ERRA   0121     ERRB   0131     EXEC   0097     EXIT   010F
EXITA  0118     FSONE  01DA     FSPC   00CF     FSPSW  00C5     FUTRY  0149
GEST   00A8     GO     019E     GSTAR  0180     H      0133     ILSGC  00F0
INIT   00B2     INITA  00B4     INSTA  00AD     INSTB  00AF     LONG   017D
LORSH  0177     M      015A     MAGT   0000     MINT   0004     MTBEN  0086
MTBSY  00A1     MTCCS  00C7     MTCMN  00B1     MTCSS  0039     MTECD  00C6
MTEOF  015D     MTFMX  00A4     MTFUN  00D4     MTGO   004B     MTIEN  0089
MTILL  0063     MTINT  00D5     MTLP   0059     MTMK2  00BF     MTMK3  00C0
MTMK6  00C1     MTMK7  00A6     MTNR   0060     MTRD   006E     MTRET  0017
MTRGO  00F2     MTRRR  00D7     MTSAV  014C     MTSV   009A     MTUST  00B0
MTWEN  0073     MTWOT  0125     MTWSV  00A5     MTW2   01AD     MTOOF  00A7
MT003  00D1     MT004  00BD     MT006  00BC     M0050  00D0     M1     00AC
M10    00C2     NOER   011D     NOTER  012A     OVER   00EC     READ   00FB
RECNT  00AA     REDY   0047     RERE   0157     RETRY  01AF     RSP    01BF
RTST   01D2     RWREI  0170     RWRSW  00D2     RWTM   0166     RWU    0199
RWUC   00CB     RWURE  01A2     RWUT   018F     SKP    00DC     T      01D5
TEMP   0114     TENSE  00DD     TMC    00CD     TSCSW  00A3     TSDAT  00B7
TSRET  00A2     TSSEA  00BA     WARET  003B     WOWTM  0128     WRT    0183
WSP    018B     WTM    01A4     WTMRE  01AA     WWOR   011B

NO ERRORS IN ABOVE ASSEMBLY.
```

```
// JOB
// ASM
*LIST    7-22.
0000 0  0000          SPACE DC        0
0001 20 176558F1            LIBF      PRNT1
0002 0  3100                DC        /3100
0003 20 176558F1            LIBF      PRNT1
0004 0  0000                DC        0
0005 0  70FD                MDX       *-3
0006 01 4C800000            BSC   I   SPACE
0008 0  40F7          BEGIN BSI       SPACE
0009 0  6105          RD    LDX    1  5
000A 20 03059130            LIBF      CARD0
000B 0  1000                DC        /1000               READ
000C 1  0153                DC        INPUT
000D 20 225C5144            LIBF      SPEED
000E 0  0000                DC        /0000               CARD TO EBCDIC CODE
000F 1  0154                DC        INPUT+1             CARD AREA
0010 1  019D                DC        INPTA+1             EBCDIC CODE AREA
0011 0  0048                DC        72                  CHARACTER CNT
0012 20 03059130            LIBF      CARD0
0013 0  0000                DC        0
0014 0  70FD                MDX       *-3
0015 20 140478C0            LIBF      MAGT
0016 0  2000                DC        /2000               WRITE ON ZR
0017 1  019C                DC        INPTA
0018 1  00F5                DC        ERRTP
0019 20 140478C0            LIBF      MAGT
001A 0  0000                DC        0
001B 0  70FD                MDX       *-3
001C 0  71FF                MDX    1  -1
001D 0  70EC                MDX       RD+1
001E 0  406D                BSI       WTM0
001F 0  406C                BSI       WTM0
0020 0  4071                BSI       RWD0
0021 0  6105                LDX    1  5
0022 20 140478C0      TRAN  LIBF      MAGT
0023 0  1000                DC        /1000
0024 1  019C                DC        INPTA
0025 1  00F5                DC        ERRTP
0026 20 140478C0            LIBF      MAGT
0027 0  2001                DC        /2001
0028 1  019C                DC        INPTA
0029 1  00F5                DC        ERRTP
002A 20 140478C0            LIBF      MAGT
002B 0  0000                DC        0
002C 0  70FD                MDX       *-3
002D 0  71FF                MDX    1  -1
002E 0  70F3                MDX       TRAN
002F 20 140478C0            LIBF      MAGT
0030 0  1000                DC        /1000
0031 1  019C                DC        INPTA
0032 1  00F5                DC        EOTSK
0033 0  4063                BSI       WTM1
0034 0  4062                BSI       WTM1
0035 0  4067                BSI       RWD1
0036 0  6105                LDX    1  5
0037 20 140478C0      PRN   LIBF      MAGT
0038 0  1001                DC        /1001
0039 1  019C                DC        INPTA
003A 1  00F5                DC        ERRTP
003B 0  406B                BSI       PRNT
003C 0  71FF                MDX    1  -1
003D 0  70F9                MDX       PRN
```

```
003E 0  4063                  BSI      BKSP1
003F 0  4062                  BSI      BKSP1
0040 0  4061                  BSI      BKSP1
0041 0  4060                  BSI      BKSP1
0042 0  405F                  BSI      BKSP1
0043 0  40BC                  BSI      SPACE
0044 0  6105                  LDX    1 5
0045 20 140478C0      RPD     LIBF     MAGT
0046 0  1001                  DC       /1001
0047 1  019C                  DC       INPTA
0048 1  00F5                  DC       EOTSK
0049 0  405D                  BSI      PRNT
004A 0  71FF                  MDX    1 -1
004B 0  70F9                  MDX      RPD
004C 20 140478C0              LIBF     MAGT
004D 0  1001                  DC       /1001
004E 1  019C                  DC       INPTA
004F 1  00F5                  DC       EOTSK
0050 0  6105                  LDX    1 5
0051 20 140478C0      PRO     LIBF     MAGT
0052 0  1000                  DC       /1000
0053 1  019C                  DC       INPTA
0054 1  00F5                  DC       ERRTP
0055 0  71FF                  MDX    1 -1
0056 0  70FA                  MDX      PRO
0057 20 140478C0              LIBF     MAGT
0058 0  1000                  DC       /1000
0059 1  019C                  DC       INPTA
005A 1  00D8                  DC       ETERM
005B 20 140478C0              LIBF     MAGT
005C 0  2000                  DC       /2000
005D 1  019C                  DC       INPTA
005E 1  00F5                  DC       ERRTP
005F 20 140478C0              LIBF     MAGT
0060 0  6000                  DC       /6000
0061 20 140478C0              LIBF     MAGT
0062 0  6000                  DC       /6000
0063 20 140478C0              LIBF     MAGT
0064 0  6000                  DC       /6000
0065 20 140478C0              LIBF     MAGT
0066 0  1000                  DC       /1000
0067 1  019C                  DC       INPTA
0068 1  00E2                  DC       REINT
0069 0  4028                  BSI      RWDO
006A 0  4095                  BSI      SPACE
006B 0  610B                  LDX    1 11
006C 20 140478C0      LAST    LIBF     MAGT
006D 0  1000                  DC       /1000
006E 1  019C                  DC       INPTA
006F 1  00E7                  DC       RWURE
0070 0  4036                  BSI      PRNT
0071 0  71FF                  MDX    1 -1
0072 0  70F9                  MDX      LAST
0073 0  408C                  BSI      SPACE
0074 20 140478C0              LIBF     MAGT
0075 0  1000                  DC       /1000
0076 1  0106                  DC       BLKLW
0077 1  00CE                  DC       ERLOW
0078 0  403B                  BSI      PRNLO
0079 20 140478C0              LIBF     MAGT
007A 0  1000                  DC       /1000
007B 1  0120                  DC       BLKHI
```

```
007C 1   00F5                   DC        ERRHI
007D 0   4043                   BSI       PRNHI
007E 0   6103                   LDX     1 3
007F 20  140478C0       SKIP    LIBF      MAGT
0080 0   1000                   DC        /1000
0081 1   0120                   DC        BLKHI
0082 1   00CE                   DC        ERLOW
0083 0   403D                   BSI       PRNHI
0084 0   71FF                   MDX     1 -1
0085 0   70F9                   MDX       SKIP
0086 20  140478C0               LIBF      MAGT
0087 0   5000                   DC        /5000
0088 20  140478C0               LIBF      MAGT
0089 0   0000                   DC        0
008A 0   70FD                   MDX       *-3
008B 0   6038                   EXIT
008C 0   0000           WTMO    DC        0
008D 20  140478C0               LIBF      MAGT
008E 0   7000                   DC        /7000
008F 1   00F5                   DC        ERRTP
0090 01  4C80008C               BSC     I WTMO
0092 0   0000           RWDO    DC        0
0093 20  140478C0               LIBF      MAGT
0094 0   4000                   DC        /4000
0095 01  4C800092               BSC     I RWDO
0097 0   0000           WTM1    DC        0
0098 20  140478C0               LIBF      MAGT
0099 0   7001                   DC        /7001
009A 1   00F5                   DC        ERRTP
009B 01  4C800097               BSC     I WTM1
009D 0   0000           RWD1    DC        0
009E 20  140478C0               LIBF      MAGT
009F 0   4001                   DC        /4001
00A0 01  4C80009D               BSC     I RWD1
00A2 0   0000           BKSP1   DC        0
00A3 20  140478C0               LIBF      MAGT
00A4 0   6001                   DC        /6001
00A5 01  4C8000A2               BSC     I BKSP1
00A7 0   0000           PRNT    DC        0
00A8 20  140478C0               LIBF      MAGT
00A9 0   0000                   DC        /0000
00AA 0   70FD                   MDX       *-3
00AB 20  176558F1               LIBF      PRNT1
00AC 0   2000                   DC        /2000
00AD 1   019C                   DC        INPTA
00AE 1   00F5                   DC        ERR
00AF 20  176558F1               LIBF      PRNT1
00B0 0   0000                   DC        0
00B1 0   70FD                   MDX       *-3
00B2 01  4C8000A7               BSC     I PRNT
00B4 0   0000           PRNLO   DC        0
00B5 20  140478C0               LIBF      MAGT
00B6 0   0000                   DC        0
00B7 0   70FD                   MDX       *-3
00B8 20  176558F1               LIBF      PRNT1
00B9 0   2000                   DC        /2000
00BA 1   0106                   DC        BLKLW
00BB 1   00F5                   DC        ERR
00BC 20  176558F1               LIBF      PRNT1
00BD 0   0000                   DC        0
00BE 0   70FD                   MDX       *-3
00BF 01  4C8000B4               BSC     I PRNLO
```

```
00C1 0  0000           PRNHI  DC        0
00C2 20 140478C0              LIBF      MAGT
00C3 0  0000                  DC        0
00C4 0  70FD                  MDX       *-3
00C5 20 176558F1              LIBF      PRNT1
00C6 0  2000                  DC        /2000
00C7 1  0120                  DC        BLKHI
00C8 1  00F5                  DC        ERR
00C9 20 176558F1              LIBF      PRNT1
00CA 0  0000                  DC        0
00CB 0  70FD                  MDX       *-3
00CC 01 4C8000C1              BSC    I  PRNHI
00CE 0  0000           ERLOW  DC        0
00CF 0  4029                  BSI       ERRCK
00D0 0  7000                  MDX       *
00D1 01 740400D0              MDX    L  *-3,+4
00D3 01 4C8000CE              BSC    I  ERLOW
00D5 0  1010                  SLA       16
00D6 01 4C8000CE              BSC    I  ERLOW
00D8 0  0000           ETERM  DC        0
00D9 0  401F                  BSI       ERRCK
00DA 0  7000                  MDX       *
00DB 01 740400DA              MDX    L  *-3,+4
00DD 01 4C8000D8              BSC    I  ETERM
00DF 0  1010                  SLA       16
00E0 01 4C8000D8              BSC    I  ETERM
00E2 0  0000           REINT  DC        0
00E3 0  4015                  BSI       ERRCK
00E4 0  1801                  SRA       1
00E5 01 4C8000E2              BSC    I  REINT
00E7 0  0000           RWURE  DC        0
00E8 0  4010                  BSI       ERRCK
00E9 0  100D                  SLA       13
00EA 01 4C2800EE              BSC    L  EOT,+Z
00EC 01 4C8000E7              BSC    I  RWURE
00EE 0  7000           EOT    MDX       *
00EF 01 740400EE              MDX    L  EOT,+4
00F1 01 4C8000E7              BSC    I  RWURE
00F3 0  1801                  SRA       1
00F4 0  70FC                  MDX       *-4
00F5 0  0000           ERRTP  DC        0                  RETRY ONLY
00F6 0  4002                  BSI       ERRCK
00F7 01 4C8000F5              BSC    I  ERRTP
00F9 0  0000           ERRCK  DC        0
00FA 0  E00A                  AND       FOFF
00FB 0  9007                  S         ONE
00FC 01 4C200100              BSC    L  NO,Z
00FE 20 17064885              LIBF      PAUSE
00FF 1  0104                  DC        DEAD
0100 0  8002           NO     A         ONE
0101 01 4C8000F9              BSC    I  ERRCK
0103 0  0001           ONE    DC        1
0104 0  DEAD           DEAD   DC        /DEAD
0105 0  FOFF           FOFF   DC        /FOFF
0106 0  0019           BLKLW  DC        25
0107    0019                  BSS       25
0120 0  0032           BLKHI  DC        50
0121    0032                  BSS       50
0153 0  0048           INPUT  DC        72                 COLUMN CNT
0154    0048                  BSS       72
019C 0  0024           INPTA  DC        36                 WORD CNT
019D    0024                  BSS       36
```

```
00F5                          ERRHI  EQU     ERRTP
00F5                          EOTSK  EQU     ERRTP
00F5                          ERR    EQU     ERRTP
01C2      0008                       END     BEGIN

     NO ERRORS IN ABOVE ASSEMBLY.
```

```
// XEQ TESTM
```

7-22A.

THIS PROGRAM TESTS THE MAGT SUBROUTINE FOR MAGNETIC TAPE I/O FOR THE IBM 1130. FIVE CARDS ARE READ AND STORED ON TAPE UNIT 0, ARE TRANSFERED TO UNIT 1, AND ARE THEN PRINTED. UNIT 1 IS THEN BACKSPACED AND THE RECORDS ARE RE-READ. FINALLY, A TEST OF THE EOT-ON-READ RECOVERY CHOICES AND THE INCORRECT LENGTH RECOVERY CHOICES ARE TESTED ON TAPE UNIT 0.

```
THIS PROGRAM TESTS THE MAGT SUBROUTINE FOR MAGNETIC TAPE I/O FOR THE IBM
1130.  FIVE CARDS ARE READ AND STORED ON TAPE UNIT 0, ARE TRANSFERED TO
UNIT 1, AND ARE THEN PRINTED.  UNIT 1 IS THEN BACKSPACED AND THE RECORDS
ARE RE-READ.  FINALLY, A TEST OF THE EOT-ON-READ RECOVERY CHOICES AND
THE INCORRECT LENGTH RECOVERY CHOICES ARE TESTED ON TAPE UNIT 0.
```

```
THIS PROGRAM TESTS THE MAGT SUBROUTINE FOR MAGNETIC TAPE I/O FOR THE IBM
1130.  FIVE CARDS ARE READ AND STORED ON TAPE UNIT 0, ARE TRANSFERED TO
UNIT 1, AND ARE THEN PRINTED.  UNIT 1 IS THEN BACKSPACED AND THE RECORDS
ARE RE-READ.  FINALLY, A TEST OF THE EOT-ON-READ RECOVERY CHOICES AND
THE INCORRECT LENGTH RECOVERY CHOICES ARE TESTED ON TAPE UNIT 0.
THIS PROGRAM TESTS THE MAGT SUBROUTINE FOR MAGNETIC TAPE I/O FOR THE IBM
1130.  FIVE CARDS ARE READ AND STORED ON TAPE UNIT 0, ARE TRANSFERED TO
UNIT 1, AND ARE THEN PRINTED.  UNIT 1 IS THEN BACKSPACED AND THE RECORDS
ARE RE-READ.  FINALLY, A TEST OF THE EOT-ON-READ RECOVERY CHOICES AND
THE INCORRECT LENGTH RECOVERY CHOICES ARE TESTED ON TAPE UNIT 0.
THE INCORRECT LENGTH RECOVERY CHOICES ARE TESTED ON TAPE UNIT 0.
```

```
THIS PROGRAM TESTS THE MAGT S BRO TINE FOR MAGNETI
1130.  FIVE CARDS ARE READ AND STORED ON TAPE UNIT 0, ARE TRANSFERED TO
UNIT 1, AND ARE THEN PRINTED.  UNIT 1 IS THEN BACKSPACED AND THE RECORDS
ARE RE-READ.  FINALLY, A TEST OF THE EOT-ON-READ RECOVERY CHOICES AND
THE INCORRECT LENGTH RECOVERY CHOICES ARE TESTED ON TAPE UNIT 0.
```

```
// JOB
// ASM
*LIST  7-23.
*LEVEL 4
                                 ILS   04
0000 0   0438         ADDR4  DC          /0438
0001 0   0734                DC          /0734
0002 0   0435                DC          /0435
0003 0   0436                DC          /0436
0004 0   0000         ILS04  DC          0
0005 0   D812                STD         TEMP4
0006 0   280C                STS         NT46
0007 0   690A                STX      1  NT44+1
0008 0   6104         NT42   LDX      1  4
0009 0   0810                XIO         SENS4-1
000A 0   1140                SLCA     1  0
000B 01  C500001E            LD      L1  DEVC4
000D 01  4C180023            BSC     L   SCTST,+-
000F 01  4580FFFF            BSI     I1  ADDR4-1
0011 00  65000000     NT44   LDX     L1  0
0013 0   2000         NT46   LDS         0
0014 0   C803                LDD         TEMP4
0015 01  4CC00004            BOSC    I   ILS04
0018     0002         TEMP4  BSS     E   2
001A 0   0000                DC          0
001B 0   0300         SENS4  DC          /0300
001C 0   0000                DC          0
001D 0   DB00         INST   DC          /DB00
001E 0   0000         DEVC4  DC          0
001F 0   0000                DC          0
0020 0   1701                DC          /1701
0021 0   0F01                DC          /0F01
0022 0   1F01                DC          /1F01
0023 0   08F8         SCTST  XIO         INST-1
0024 0   100C                SLA         12
0025 01  4C100011            BSC     L   NT44,-
0027 01  44800000            BSI     I   ADDR4
0029 0   70F7                MDX         NT44
002A                         END

     NO ERRORS IN ABOVE ASSEMBLY.
```

```
// JOB
// ASM
*LIST    7-24.
*PRINT SYMBOL TABLE
                                     LIBR
0000       140478E9                  ENT        MAGTZ
0000 0    7005              MAGTZ   MDX        ENTRY              ISS CALL ENTRY
0001 00   4CC00000          EXIT    BOSC  I    *-*                CALL EXIT
0003 0    0032              C100    DC         50                 READ RETRY COUNT
0004 0    0003              C003    DC         3                  WRITE/WTM RETRY  CNT
0005 0    0000              AREA    DC         0                  SAVE
003C                        IOBUF   EQU        60
0006 01   650000F1          ENTRY   LDX   L1   EXINT              SET INTER ADDR
0008 00   6D00000C                  STX   L1   12
000A 0    613C                      LDX    1   IOBUF
000B 0    907D                      S          C002
000C 0    D07D                      STO        RDWRT              SAVE OP CODE
000D 01   4C280013                  BSC   L    *+4,+Z             IF READ, BRANCH
000F 0    1010                      SLA        16                  IF NT READ, SET EOTSW OFF
0010 01   D40000B3                  STO   L    EOTSW
0012 0    C077                      LD         RDWRT
0013 0    4808                      BSC        +                TETEST FOR RD/W  IF NT SKP
0014 0    C87D                      LDD        UNIT-1             IF RD/W, USE OLD UNIT
0015 0    1090                      SLT        16
0016 01   940000A4                  S     L    IOCC2
0018 01   4C3000ED                  BSC   L    PAT,Z-             IF NO-OP, BRANCH
001A 01   840000A4                  A     L    IOCC2
001C 0    D076                   [  STO        UNIT               RESET UNIT
001D 0    E86E                 A [  OR         EOFO               FORM EOFX
001E 0    D06D                   [  STO        EOFD                   AND STORE
001F 0    C079                      LD         IOCC+1
0020 0    E06F                      AND        FF00
0021 0    E871                      OR         UNIT               IOCC DEVICE
0022 0    E86D                      OR         0080
0023 0    D075                      STO        IOCC+1                 SET UP
0024 0    D072                      STO        TSSEN+1
0025 0    D079                      STO        SDATA+1
0026 0    623D                      LDX    2   61                 SET COUNT
0027 0    6ADD                      STX    2   AREA
0028 0    C061                      LD         RDWRT              LOAD OP CODE
0029 01   4C280042                  BSC   L    READ,+Z            READ
002B 01   4C180048                  BSC   L    WRIT,+-            WRITE
002D 0    905A                      S          C001
002F 01   4C180037                  BSC   L    REWD,+-            REWIND
0030 0    9057                      S          C001
0031 01   4C180039                  BSC   L    BSPC,+-            BACKSPACE
0033 0    1010                      SLA        16                 SET RDWRT TO WRITE FOR WT
0034 0    D055                      STO        RDWRT                    RETRIES
0035 0    C073                      LD         CEOF               END OF FILE
0036 0    7021                      MDX        ENTIO
0037 0    C06C              REWD    LD         CREWD
0038 0    7001                      MDX        BSPC+1
0039 0    C06B              BSPC    LD         CBSPC
003A 0    1890                      SRT        16
003B 0    403B                      BSI        TREDY              TEST DEV RDY
003C 0    C073                      LD         DATA
003D 0    1803                      SRA        3                  SET LP MARKER
003E 01   4C040001                  BSC   L    EXIT,E             EXIT IF ON
0040 0    1090                      SLT        16
0041 0    7016                      MDX        ENTIO
0042 0    C0C0              READ    LD         C100               READ
0043 0    D047                      STO        ERTST              SET RETRY COUNTER
0044 0    C04D                      LD         OCNT
0045 0    D100                      STO    1   0                  SET WORD COUNT
```

```
0046 0   C055                   LD          CREAD
0047 0   7010                   MDX         ENTIO
0048 0   C0BB          WRIT     LD          C003            WRITE
0049 0   D041            B →    STO         ERTST
004A 0   623C                   LDX     2   IOBUF           PACK BUFFER FOR OUTPT
004B 0   7102          LOOP1    MDX     1   2
004C 0   7201                   MDX     2   1
004D 0   C1FE                   LD      1   -2
004E 0   1008                   SLA         8
004F 0   E9FF                   OR      1   -1
0050 0   D200                   STO     2   0
0051 01  74FF0005               MDX     L   AREA,-1
0053 0   70F7                   MDX         LOOP1
0054 0   C03D                   LD          OCNT
0055 00  D400003C               STO     L   IOBUF
0057 0   C04F                   LD          CWRIT
0058 0   D05B          ENTIO    STO         HOLD
0059 0   1010          IOOPA    SLA         16
005A 0   D03F                   STO         ERCNT           INIT ERROR CNT
005B 0   C059          IOOPB    LD          HOLD            LOAD COMMAND
005C 0   D051                   STO         CCW+1           SET COOMAND INTO CCW
005D 0   10A0          IOOP     SLT         32
005E 0   D835                   STD         ERSW            CLEAR ERROR SWITCH
005F 0   4075                   BSI         TNRDY            EXEC OP AND AWAIT INTER
0060 0   C033                   LD          ERSW
0061 01  4C20008B               BSC     L   ERROR,Z         BRANCH IF ERROR
0063 0   C026          ENTEF    LD          RDWRT
0064 01  4C100081               BSC     L   EXIT,-          EXIT IF NOT READ
0066 0   1010                   SLA         16
0067 0   D04B                   STO         EOTSW            SET SWT TO OFF
0068 0   6278                   LDX     2   120             UNPACK INPUT
0069 0   6178                   LDX     1   IOBUF+60
006A 0   C101                   LD      1   1
006B 0   1808                   SRA         8
006C 0   D23C                   STO     2   IOBUF
006D 0   C100          LOOP2    LD      1   0
006E 0   18C8                   RTE         8
006F 0   D23A                   STO     2   IOBUF-2
0070 0   1010                   SLA         16
0071 0   1088                   SLT         8
0072 0   D23B                   STO     2   IOBUF-1
0073 0   71FF                   MDX     1   -1
0074 0   72FE                   MDX     2   -2
0075 0   70F7                   MDX         LOOP2
0076 0   708A          EXITA    MDX         EXIT
0077 0   0000          TREDY    DC          0               TEST UNIT READY ,NT BSY
0078 0   1010                   SLA         16
0079 0   D02E                   STO         NBSW            SET INTER SWT TO OFF
007A 0   0823                   XIO         SDATA           FETCH SENSE DATA
007B 0   4061                   BSI         WAIT             AWAIT INTER
007C 0   C033                   LD          DATA
007D 0   100A                   SLA         10              SET TUA, TUB BITS
007E 01  4C020083               BSC     L   REDY,C          IF READY, BRANCH
0080 20  17054885               LIBF        PAUSE           IF NT RDY, INDICATE
0081 1   0087                   DC          FRADA
0082 0   70F5                   MDX         TREDY+1         RETEST
0083 01  4C680078      REDY     BOSC    L   TREDY+1,+Z      IF BUSY, RETEST
0085 01  4C800077               BSC     I   TREDY           IF READY, GO
0087 0   DEAD          FRADA    DC          /DEAD
0088 0   0001          C001     DC          1
0089 0   0002          C002     DC          2
008A 0   0000          RDWRT    DC          0
```

```
008B 0   0000           ERTST  DC       0
008C 0   0000           EOFD   DC       0
008D 0   E0F0           EOFO   DC       /E0F0
008E 0   FFC7           WCTST  DC       -57
008F 0   FF00           FF00   DC       /FF00
0090 0   0080           0080   DC       /0080
0092     0000                  BSS   E  0
0092 0   403D           OCNT   DC       /403D
0093 0   0000           UNIT   DC       0
0094 0   0000           ERSW   DC       0
0095 0   0000           NOISE  DC       0
0096 1   00AA           TSSEN  DC       CCWA
0097 0   DD00                  DC       /DD00
0098 1   00AD           IOCC   DC       CCW              START
0099 0   DD00                  DC       /DD00               I/O
009A 0   0000           SENSE  DC       0                SENSE U STAT W/ RESET
009B 0   DF03                  DC       /DF03
009C 0   2002           SNSWC  DC       /2002            READ
009D 0   DF06                  DC       /DF06            SENSE BYTE CNT
009E 1   00A0           SDATA  DC       SDATA+2
009F 0   DD00                  DC       /DD00
00A0 0   0006                  DC       6
00A1 0   0004                  DC       4
00A2 1   00B0                  DC       DATA
00A3 0   0017           IOCC1  DC       /0017            ERASE
00A4 0   0007           IOCC2  DC       /0007            REWIND
00A5 0   0027           CBSPC  DC       /0027            BACKSPACE
00A6 0   2001           IOCC3  DC       /2001            WRITE
00A7 0   000F                  DC       /000F            RFU
00A8 0   0000           WRSW   DC       0
00A9 0   201F           CEOF   DC       /201F            WTM
009C                    CREAD  EQU      SNSWC
00A6                    CWRIT  EQU      IOCC3
00A3                    CERAS  EQU      IOCC1
00A4                    CREWD  EQU      IOCC2
009A                    ERCNT  EQU      SENSE
00AA 0   0000           CCWA   DC       0                TEST I/O(CCW)
00AB 0   0000                  DC       0
00AC 0   0000                  DC       0
00AD 0   007A           CCW    DC       122              BYTES
00AE 0   0000                  DC       0                COMMAND
00AF 0   003D                  DC       /003D            IOBUF+1 ADDR
00B0     0003           DATA   BSS      3
00B3 0   0000           EOTSW  DC       0
00B4 0   0000           HOLD   DC       0
00B5 0   C0F3           TYECT  LD       CEOF             WTM
00B6 0   D0F7                  STO      CCW+1
00B7 01  4C4000B9              BOSC  L  *
00B9 0   401F                  BSI      TNRDY
00BA 0   70BB                  MDX      EXITA
00BB 0   C0CF           ERROR  LD       RDWRT
00BC 01  4C300001              BSC   L  EXIT,-Z          EXIT IF NT RD/WRT
00BE 01  4C2000E3              BSC   L  CKNOS,Z          BRANCH IF READ
00C0 0   C0E4                  LD       CBSPC              (WRITE ERROR)
00C1 0   D0FC                  STO      CCW+1
00C2 0   4012                  BSI      TNRDY            BACK SPACE
00C3 0   C0DF                  LD       CERAS            SET ERASE
00C4 0   D0F9                  STO      CCW+1
00C5 0   400F           ERDWT  BSI      TNRDY            EXEC ERASE OR BSP
00C6 0   C0D3                  LD       ERCNT
00C7 0   80C0                  A        C001             INCRM ERR CNT
00C8 0   D0D1                  STO      ERCNT
```

```
00C9 0   90C1                S        ERTST              CNT OVER MAX
00CA 01  4C08005B            BSC   L  IOOPB,+            NO, RETRY OPERATION
00CC 0   C0E1                LD       CCW+1
00CD 0   F0DB                EOR      CFOF               TEST FOR WTM
00CE 01  4C2000E8            BSC   L  PERM,Z             IF NT WTM, INDIC. PERM. ERR
00D0 01  740000B3            MDX   L  EOTSW,0
00D2 0   70E2                MDX      TMEOT              IF EOT, RETRY
00D3 0   D0C6                STO      ERCNT
00D4 0   70BB                MDX      IOOP               IF WTM, RETRY
00D5 0   0000       TNRDY    DC       0
00D6 0   40A0                BSI      TREDY              UNIT READY
00D7 0   1010                SLA      16
00D8 0   D0CF                STO      NBSW
00D9 0   08BE                XIO      IOCC               EXECUTE OP
00DA 0   4002                BSI      WAIT               AWAIT INTER
00DB 01  4C8000D5            BSC   I  TNRDY              RET AFTER INTER
00DD 0   0000       WAIT     DC       0
00DE 0   C0C9                LD       NBSW
00DF 01  4C1800DE            BSC   L  WAIT+1,+-           IF NO INTER YET, WAIT
00E1 01  4C8000DD            BSC   I  WAIT               RETURN AFTER INTER
00E3 0   C0B1       CKNOS    LD       NOISE
00E4 01  4C200059            BSC   L  IOOPA,Z            SKIP NOISE RECORD
00E6 0   C0BF                LD       CBSPC              BACKSPACE
00E7 0   70DC                MDX      ERDWT-1
00E8 20  17064885   PERM     LIBF     PAUSE               IF ERR, INDICATE
00E9 1   00EC                DC       ERAD
00EA 01  4C000063            BSC   L  ENTEF              CONTIN & EXIT IF RETURNED
00EC 0   BAD0       ERAD     DC       /BAD0
00ED 0   E0B9       PAT      AND      IOCC3+1
00EE 0   9099                S        C001
00EF 0   D0A3                STO      UNIT                SET NEW UNIT
00F0 0   70C9                MDX      ERROR-1
00F1 0   0000       EXINT    DC       0                  ISS INTER RET LINK
00F2 0   08A9       INTRP    XIO      SNSWC              IOCC BYTE SENSE
00F3 0   909A                S        WCTST              CHK NOISE
00F4 0   4828                BSC      +Z
00F5 0   D09F                STO      NOISE
00F6 0   08A3                XIO      SENSE              UNIT STAT, RESET
00F7 0   100D                SLA      13                  SET DE
00F8 01  4C100112            BSC   L  OUTIN,-             IF DE NT ON, AWAIT SECND INT.
00FA 0   1001                SLA      1                   SET UC BIT
00FB 0   4828                BSC      +Z
00FC 0   6897                STX      ERSW               SET ERSW NON ZERO
00FD 0   68AA                STX      NBSW               SET NBSW NON ZERO
00FE 0   1001                SLA      1                  SET UE(EOT,EOF)
00FF 01  4C100112            BSC   L  OUTIN,-            IF NT ON, EXIT
0101 0   C0AC                LD       CCW+1
0102 0   90A3                S        IOCC3
0103 01  4C180114            BSC   L  WTEOR,+-           IF WRITE, WTM(2)
0105 01  4C080112            BSC   L  OUTIN,+            IF NT READ, EXIT
0107 01  740000B3            MDX   L  EOTSW,0            IF READ, IS EOT ON
0109 0   7006                MDX      RWU                IF YES, RWU/TERM
010A 01  740300B3            MDX   L  EOTSW,+3           IF NT ON, SET ON
010C 20  17064885            LIBF     PAUSE              EOF INDICATE
010D 1   008C                DC       EOFD
010E 01  4C400059            BOSC  L  IOOPA
0110 0   C096       RWU      LD       IOCC3+1
0111 0   70A4                MDX      TMEOT+1             EXEC RWU/TERM
0112 01  4CC000F1   OUTIN    BOSC  I  EXINT
0114 01  74FF00C4   WTEOR    MDX   L  C003,-1
0116 0   709F                MDX      TMEOT
0117 01  74030004            MDX   L  C003,+3
```

```
0119 0   70F6                  MDX        RWU
011A                           END
```

```
                                     SYMBOL  TABLE

   AREA   0005      BSPC   0039      CBSPC  00A5      CCW    00AD      CCWA   00AA
   CEOF   00A9      CERAS  00A3      CKNOS  00E3      CREAD  009C      CREWD  00A4
   CWRIT  00A6      C001   0088      C002   0089      C003   0004      C100   0003
   DATA   00B0      ENTEF  0063      ENTIO  0058      ENTRY  0006      EOFD   008C
   EOFO   008D      ECTSW  00B3      ERCNT  009A      ERDWT  00C5      ERROR  00BB
   FRSW   0094      ERTST  008B      EXINT  00F1      EXIT   0001      EXITA  0076
   FBAD   00FC      FBADA  0087      FF00   008F      HOLD   00B4      INTRP  00F2
 Λ IOBUF  003C      IOCC   0098      IOCC1  00A3      IOCC2  00A4      IOCC3  00A6
   IOOP   005D      IOOPA  0059      IOOPB  005B      LOOP1  004B      LOOP2  006D
   MAGTZ  0000      NRSW   00A8      NOISE  0095      OCNT   0092      OC80   0090
   OUTIN  0112      PAT    00ED      PERM   00E8      RDWRT  008A      READ   0042
   REDY   0083      REWD   0037      RWU    0110      SDATA  009E      SENSE  009A
   SNSWC  009C      TMEOT  00B5      TNRDY  00D5      TREDY  0077      TSSEN  0096
   UNIT   0093      WAIT   00DD      WCTST  008E      WRIT   0048      WTEOR  0114

   NO  ERRORS  IN  ABOVE  ASSEMBLY.
```

```
// JOB
// FOR
*LISTALL   7-25.
*NAME TAPEF
*IOCS(CARD,MAGNETIC TAPE,1132 PRINTER)
       DIMENSION X(20)
       END FILE 8
       DO 5 K=1,9
       K=K+1
       READ(2,1)(X(I),I=1,18)
 5     WRITE(5,1)(X(I),I=1,18)
 1     FORMAT(18A4)
       END FILE 0
       END FILE 0
       REWIND 0
       DO 10 K=1,11
       K=K+1
       REWIND 8
       READ(5,1)(X(I),I=1,18)
       REWIND 9
 10    WRITE(5,1)(X(I),I=1,18)
       END FILE 1
       END FILE 1
       REWIND 1
       REWIND 9
       DO 15 K=1,13
       K=K+1
       READ(5,1)(X(I),I=1,18)
 15    WRITE(3,1)(X(I),I=1,18)
       CALL EXIT
       END

VARIABLE ALLOCATIONS
 X      =0026   K      =0028   I      =002A

STATEMENT ALLOCATIONS
 1      =0038   5      =0070   10     =0089   15     =0100

FEATURES SUPPORTED
 IOCS

CALLED SUBPROGRAMS
 FLD      FSTO      SRED      SWRT      SCOMP      SFIO      SIOFX      SUBSC      EOFZ      [illegible]

INTEGER CONSTANTS
     8=002E        1=002F         9=0030         2=0031         18=0032         5=0033         0=0

CORE REQUIREMENTS FOR TAPEF
 COMMON        0   VARIABLES       46   PROGRAM      242

END OF COMPILATION
```

```
// XEQ TAPEF
```

7-25A.

THIS PROGRAM TESTS THE MAGNETIC TAPE SUPPORT FOR FORTRAN PROGRAMS ON THE IBM 1130 SYSTEM. THE TEST CONSISTS OF READING 72 COLUMNS FROM EACH OF FIVE DATA CARDS, WRITING THE CONTENTS OF EACH CARD ONTO TAPE UNIT 0, TRANSFERING THE FIVE RECORDS FROM TAPE UNIT 0 TO TAPE UNIT 1, AND FINALLY, READING THE RECORDS FROM TAPE UNIT 1 AND PRINTING THEM.

```
// JOB
// ASM
*LIST    7-26.
                                   LIBR
0000         095A4000              ENT         IOU
0000 0    900A             IOU     S           M16          IS UNIT LEGAL
0001 00   66800000                 LDX    I2   *-*
0003 0    6A06                     STX     2   RET+1
0004 01   4C100009                 BSC    L    RET,-        IF NT EXIT
0006 0    1008                     SLA         8
0007 0    E804                     OR          T0005
0008 0    E004                     AND         TOF05
0009 00   4C000000         RET     BSC    L    *-*
000B 0    0010             M16     DC          16
000C 0    0005             T0005   DC          /0005
000D 0    0F05             TOF05   DC          /0F05
000E                               END

    NO ERRORS IN ABOVE ASSEMBLY.
```

```
// JOB
// ASM
*LIST     7-27.
                                      LIBR
0001      19166569                    ENT          REWNZ
0017      020D28A9                    ENT          BCKSZ
001B      05586A40                    ENT          EOFZ
0000 0    0003              THREE     DC           3
0001 0    C0FE              REWNZ     LD           THREE
0002 00   66800000                    LDX     I2   *-*
0004 0    D01E              COM       STO          SAVAQ
0005 0    C019                        LD           H4C00
0006 0    D00E                        STO          RET
0007 0    10A0                        SLT          32
0008 00   C6800000                    LD      I2   0
000A 0    7201                        MDX      2   1
000B 0    6A0A                        STX      2   RET+1
000C 20   095A4000                    LIBF         IOU
000D 0    4808                        BSC          +
000E 0    7006                        MDX          RET
000F 0    18D8                        RTE          24
0010 0    900F                        S            H0500
0011 0    4C20                        BSC          Z
0012 0    7002                        MDX          RET
0013 0    C00F                        LD           SAVAQ
0014 20   140478E9          MAG       LIBF         MAGTZ
0015 00   4C000000          RET       BSC     L    *-*
0017 0    C00A              BCKSZ     LD           FOUR
0018 00   66800000                    LDX     I2   *-*
001A 0    70E9                        MDX          COM
001B 0    C005              EOFZ      LD           FIVE
001C 00   66800000                    LDX     I2   *-*
001E 0    70F5                        MDX          COM
001F 0    4C00              H4C00     DC           /4C00
0020 0    0500              H0500     DC           /0500
0021 0    0005              FIVE      DC           5
0022 0    0004              FOUR      DC           4
0023 0    0000              SAVAQ     DC           0
0024                                  END

     NO ERRORS IN ABOVE ASSEMBLY.
```

```
// JOB
// ASM
*LIST      7-28.
*PRINT SYMBOL TABLE
0000       140478C1              ENT         MAGTA
0000       0000           MAGTA  BSS         0
0000 0     0000           EXIT   DC          0
0001 01    660000D3       ENTRY  LDX     L2  EXINT          SET INTER, ENTRANCE ADDR
0003 00    6E00000C              STX     L2  12
0005 01    66800000              LDX     I2  MAGTA
0007 00    C6800000              LD      I2  0              COMMAND
0009 0     9065                  S           C002
000A 0     D065                  STO         RDWRT          SAVE OP CODE
000B 01    4C280010              BSC     L   *+3,+Z          IF READ, BRANCH
000D 0     1010                  SLA         16              IF NT READ, SET EOTSW OFF
000E 01    D400009B              STO     L   EOTSW
0010 00    C6800001              LD      I2  1               UNIT
0012 0     D068            A   [ STO         UNIT           RESET UNIT
0013 0     E85F                [ OR          EOFO           FORM EOFX
0014 0     D05D                [ STO         EOFD               AND STORE
0015 0     C06B                  LD          IOCC+1
0016 0     E05E                  AND         FF00
0017 0     E863                  OR          UNIT           IOCC DEVICE
0018 0     E85D                  OR          0080
0019 0     D067                  STO         IOCC+1             SET UP
001A 0     D064                  STO         TSSEN+1
001B 0     D06B                  STO         SDATA+1
001C 00    C6800002              LD      I2  2               LOAD WORD CNT
001E 0     D078                  STO         CCW+2
001F 0     1001                  SLA         1
0020 0     D074                  STO         CCW
0021 0C    C6000003              LD      L2  3               LOAD ADDR OF I/O AREA
0023 0     9073                  S           CCW+2
0024 0     804A                  A           C002
0025 0     D071                  STO         CCW+2
0026 0     C049                  LD          RDWRT          LOAD OP CODE
0027 01    4C280040              BSC     L   READ,+Z        READ
0029 01    4C180044              BSC     L   WRIT,+-        WRITE
002B 0     9042                  S           C001
002C 01    4C180035              BSC     L   REWD,+-        REWIND
002E 0     903F                  S           C001
002F 01    4C180037              BSC     L   BSPC,+-        BACKSPACE
0031 0     1010                  SLA         16             SET RDWRT TO WRITE FOR WTM
0032 0     D03D                  STO         RDWRT                 RETRIES
0033 0     C05D                  LD          CEOF           END OF FILE
0034 0     7014                  MDX         ENTIO
0035 0     C056           REWD   LD          CREWD
0036 0     7001                  MDX         BSPC+1
0037 0     C055           BSPC   LD          CBSPC
0038 0     1890                  SRT         16
0039 0     4023                  BSI         TREDY          TEST DEV RDY
003A 0     C05D                  LD          DATA
003B 0     1803                  SRA         3              SET LP MARKER
003C 01    4CC40000              BOSC    I   EXIT,E         EXIT IF ON
003E 0     1090                  SLT         16
003F 0     7009                  MDX         ENTIO
0040 0     C037           READ   LD          C100           READ
0041 0     D02F                  STO         ERTST          SET RETRY COUNTER
0042 0     C041                  LD          CREAD
0043 0     7003                  MDX         ENTIO-2
0044 0     C034           WRIT   LD          C003           WRITE
0045 0     D02B           B ---> STO         ERTST
0046 0     C047                  LD          CWRIT
0047 01    74020000              MDX     L   MAGTA,+2
```

```
0049 0   D030           ENTIO STO     HOLD
004A 01  74020000             MDX   L MAGTA,+2
004C 0   1010           IOOPA SLA     16
004D 0   D034                 STO     ERCNT               INIT ERROR CNT
004E 0   C02B           IOOPB LD      HOLD                LOAD COMMAND
004F 0   D046                 STO     CCW+1               SET COOMAND INTO CCW
0050 0   10A0           IOOP  SLT     32
0051 0   D82A                 STD     ERSW                CLEAR ERROR SWITCH
0052 0   4069                 BSI     TNRDY                EXEC OP AND AWAIT INTER
0053 0   C028                 LD      ERSW
0054 01  4C2000A2             BSC   L ERROR,Z             BRANCH IF ERROR
0056 0   C019           ENTFF LD      RDWRT
0057 01  4CD00000             BOSC  I EXIT,-              EXIT IF NOT READ
0059 0   1010                 SLA     16
005A 0   D040                 STO     EOTSW                SET SWT TO OFF
005B 01  4CC00000       EXITA BOSC  I EXIT
005D 0   0000           TREDY DC      0                   TEST UNIT READY ,NT BSY
005E 0   1010                 SLA     16
005F 0   D030                 STO     NBSW                SET INTER SWT TO OFF
0060 0   0825                 XIO     SDATA               FETCH SENSE DATA
0061 0   4062                 BSI     WAIT                 AWAIT INTER
0062 0   C035                 LD      DATA
0063 0   100A                 SLA     10                  SET TUA, TUB BITS
0064 01  4C020069             BSC   L REDY,C              IF READY, BRANCH
0066 20  17064885             LIBF    PAUSE               IF NT RDY, INDICATE
0067 1   006D                 DC      FBADA
0068 0   70F5                 MDX     TREDY+1             RETEST
0069 01  4C68005E       REDY  BOSC  L TREDY+1,+Z          IF BUSY, RETEST
006B 01  4C80005D             BSC   I TREDY               IF READY, GO
006D 0   DEAD           FBADA DC      /DEAD
006E 0   0001           C001  DC      1
006F 0   0002           C002  DC      2
0070 0   0000           RDWRT DC      C
0071 0   000C           ERTST DC      C
0072 0   0000           FOFD  DC      0
0073 0   E0F0           EOFO  DC      /EOFO
0074 0   FFF4           WCTST DC      -12
0075 0   FF00           FF00  DC      /FF0C
0076 0   0080           0080  DC      /0080
0078     0000                 BSS   E 0
0078 0   0032           C100  DC      50
0079 0   0003           C003  DC      3                   READ RETRY COUNT
007A 0   0000           HOLD  DC      0                   WRITE/WTV RETRY  CNT
007B 0   0000           UNIT  DC      0
007C 0   0000           FRSW  DC      0
007D 0   0000           NOISE DC      0
007E 1   0092           TSSFN DC      CCWA
007F 0   DD00                 DC      /DD00
0080 1   0095           IOCC  DC      CCW                 START
0081 0   DD00                 DC      /DD00                  I/O
0082 0   0000           SENSE DC      0                   SENSE U STAT W/ RESET
0083 0   DF03                 DC      /DF03
0084 0   2002           SNSWC DC      /2002               READ
0085 0   DF06                 DC      /DF06               SENSE BYTE CNT
0086 1   0088           SDATA DC      SDATA+2
0087 0   DD00                 DC      /DD00
0088 0   0006                 DC      6
0089 0   0004                 DC      4
008A 1   0098                 DC      DATA
008B 0   0017           IOCC1 DC      /0017               ERASE
008C 0   0007           IOCC2 DC      /0007               REWIND
008D 0   0027           CBSPC DC      /0027               BACKSPACE
```

```
008E 0  2001       IOCC3  DC       /2001          WRITE
008F 0  000F              DC       /000F          RWU
0090 0  000C       NBSW   DC       0
0091 0  201F       CEOF   DC       /201F          WTM
0084               CREAD  EQU      SNSWC
008E               CWRIT  EQU      IOCC3
008B               CERAS  EQU      IOCC1
008C               CREWD  EQU      IOCC2
0082               ERCNT  EQU      SENSE
0092 0  0000       CCWA   DC       0              TEST I/O(CCW)
0093 0  0000              DC       0
0094 0  0000              DC       0
0095 0  007A       CCW    DC       122            BYTES
0096 0  0000              DC       0              COMMAND
0097 0  003D              DC       /003D          IOBUF+1 ADDR
0098    0003       DATA   BSS      3
009B 0  0000       EOTSW  DC       0
009C 0  C0F4       TMEOT  LD       CEOF           WTM
009D 0  D0F8              STO      CCW+1
009E 01 4C4000A0          BOSC L   *
00A0 0  401B              BSI      TNRDY
00A1 0  70B9              MDX      EXITA
00A2 0  C0C9       ERROR  LD       RDWRT
00A3 01 4CF00000          BOSC I   EXIT,-Z        EXIT IF NT RD/WRT
00A5 01 4C2000CA          BSC  L   CKNOS,Z        BRANCH IF READ
00A7 0  C0E5              LD       CBSPC            (WRITE ERROR)
00A8 0  D0ED              STO      CCW+1
00A9 0  4012              BSI      TNRDY          BACK SPACE
00AA 0  C0E0              LD       CERAS          SET ERASE
00AB 0  D0EA              STO      CCW+1
00AC 0  400F       ERDWT  BSI      TNRDY          EXEC ERASE OR BSP
00AD 0  C0D4              LD       ERCNT
00AE 0  80BF              A        C001           INCRM ERR CNT
00AF 0  D0D2              STO      ERCNT
00B0 0  90C3              S        ERTST          CNT OVER MAX
00B1 01 4C08004E          BSC  L   IOOPB,+        NO, RETRY OPERATION
00B3 0  C0E2              LD       CCW+1
00B4 0  F0DC              EOR      CEOF           TEST FOR WTM
00B5 01 4C2000CF          BSC  L   PERM,Z         IF NT WTM, INDIC. PERM. ERR
00B7 01 7400009B          MDX  L   EOTSW,0
00B9 0  70E2              MDX      TMEOT          IF EOT, RETRY
00BA 0  D0C7              STO      ERCNT
00BB 0  7094              MDX      IOOP           IF WTM, RETRY
00BC 0  0000       TNRDY  DC       0
00BD 0  409F              BSI      TREDY          UNIT READY
00BE 0  1010              SLA      16
00BF 0  D0D0              STO      NBSW
00C0 0  08BF              XIO      IOCC           EXECUTE OP
00C1 0  4002              BSI      WAIT           AWAIT INTER
00C2 01 4C8000BC          BSC  I   TNRDY          RET AFTER INTER
00C4 0  0000       WAIT   DC       0
00C5 0  C0CA              LD       NBSW
00C6 01 4C1800C5          BSC  L   WAIT+1,+-      IF NO INTER YET, WAIT
00C8 01 4C8000C4          BSC  I   WAIT           RETURN AFTER INTER
00CA 0  C0B2       CKNOS  LD       NOISE
00CB 01 4C20004C          BSC  L   IOOPA,Z        SKIP NOISE RECORD
00CD 0  C0BF              LD       CBSPC          BACKSPACE
00CE 0  70DC              MDX      ERDWT-1
00CF 20 17054885   PERM   LIBF     PAUSE          IF ERR, INDICATE
00D0 1  0002              DC       FBAD
00D1 0  7084              MDX      ENTER          CONTINUE & EXIT IF RETURNED
00D2 0  BAD0       FBAD   DC       /BAD0
```

```
00D3 0   0000       EXINT DC        0             ISS INTER RET LINK
00D4 0   08AF       INTRP XIO       SNSWC         IOCC BYTE SENSE
00D5 0   909E             S         WCTST         CHK NOISE
00D6 0   4828             BSC       +Z
00D7 0   D0A5             STO       NOISE
00D8 0   08A9             XIO       SENSE         UNIT STAT, RESET
00D9 0   100D             SLA       13             SET DE
00DA 01  4C1000F4         BSC   L   OUTIN,-        IF DE NT ON, AWAIT SECND INT.
00DC 0   1001             SLA       1              SET UC BIT
00DD 0   4828             BSC       +Z
00DE 0   689D             STX       ERSW          SET ERSW NON ZERO
00DF 0   68B0             STX       NBSW          SET NBSW NON ZERO
00E0 0   1001             SLA       1             SET UE(EOT,EOF)
00E1 01  4C1000F4         BSC   L   OUTIN,-       IF NT ON, EXIT
00E3 0   C0B2             LD        CCW+1
00E4 0   90A9             S         IOCC3
00E5 01  4C1800F6         BSC   L   WTEOR,+-      IF WRITE, WTM(2)
00E7 01  4C0800F4         BSC   L   OUTIN,+       IF NT READ, EXIT
00E9 01  7400009B         MDX   L   EOTSW,0       IF READ, IS EOT ON
00EB 0   7006             MDX       RWU           IF YES, RWU/TERM
00EC 01  7403009B         MDX   L   EOTSW,+3      IF NT ON, SET ON
00EE 20  17064885         LIBF      PAUSE         EOF INDICATE
00EF 1   0072             DC        EOFD
00F0 01  4C40004C         BOSC  L   IOOPA
00F2 0   C09C       RWU   LD        IOCC3+1
00F3 0   70A9             MDX       TMEOT+1        EXEC RWU/TERM
00F4 01  4CC000D3   OUTIN BOSC  I   EXINT         INTER. EXIT
00F6 01  74FF0079   WTEOR MDX   L   C003,-1
00F8 0   70A3             MDX       TMEOT
00F9 01  74030079         MDX   L   C003,+3
00FB 0   70F6             MDX       RWU
00FC                      END
```

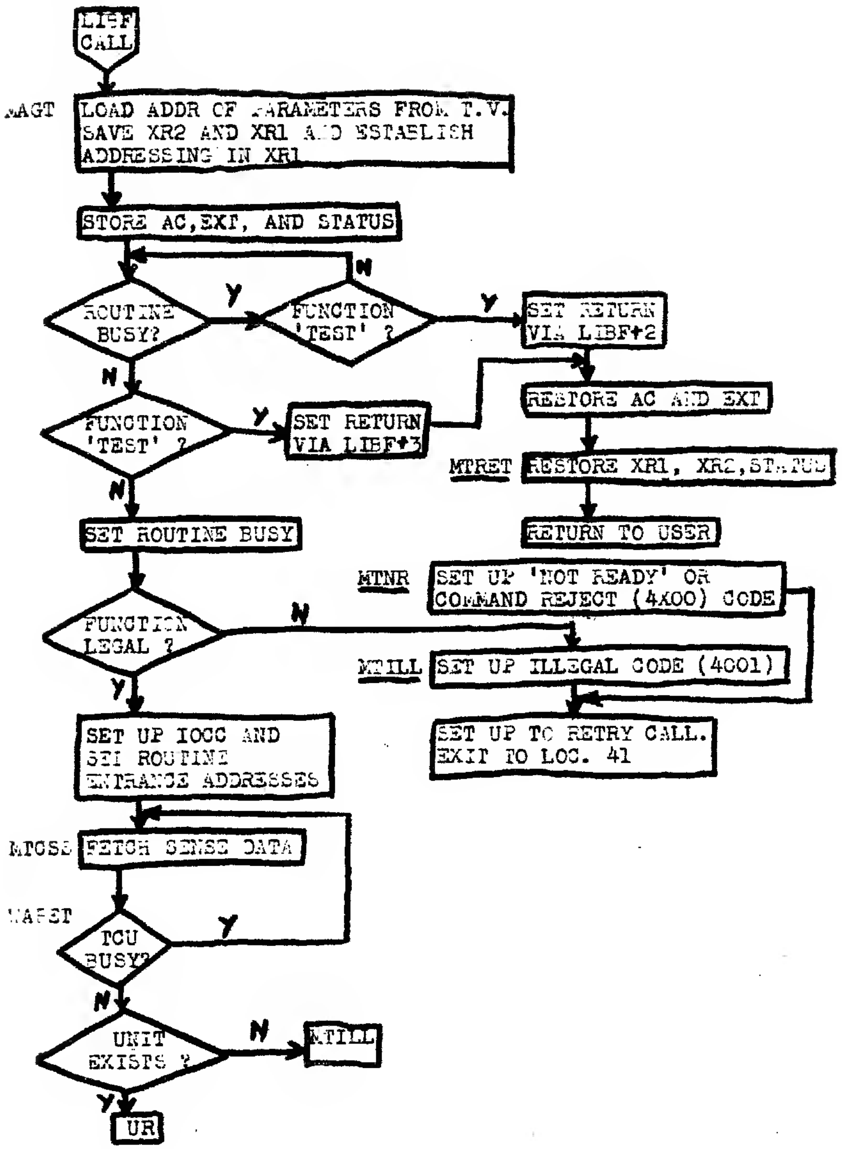
```
                                SYMBOL TABLE

BSPC   0037     CBSPC  008D     CCW    0095     CCWA   0092     CEOF   0091
CERAS  008B     CKNOS  00CA     CREAD  0084     CREWD  008C     CWRIT  008E
C001   006E     C002   006F     C003   0079     C100   007B     DATA   0098
FNTEF  0056     ENTIO  0049     ENTRY  0001     EOFD   0072     EOFO   0073
EOTSW  009B     ERCNT  0082     ERDWT  00AC     ERROR  00A2     ERSW   007C
ERTST  0071     EXINT  00D3     EXIT   0000     EXITA  005B     FBAD   00D2
FBADA  006D     FF00   0075     HOLD   007A     INTRP  00D4     IOCC   0080
IOCC1  008B     IOCC2  008C     IOCC3  008E     IOOP   0050     IOOPA  004C
IOOP3  004E     MAGTA  0000     NBSW   0090     NOISE  007D     O080   0076
OUTIN  00F4     PERM   00CF     RDWRT  0070     READ   0040     REDY   0069
REWD   0035     RWU    00F2     SDATA  0086     SENSE  0082     SNSWC  0084
TMEOT  009C     TNRDY  00BC     TREDY  005D     TSSEN  007E     UNIT   007B
WAIT   00C4     WCTST  0074     WRIT   0044     WTEOR  00F6

NO ERRORS IN ABOVE ASSEMBLY.
```

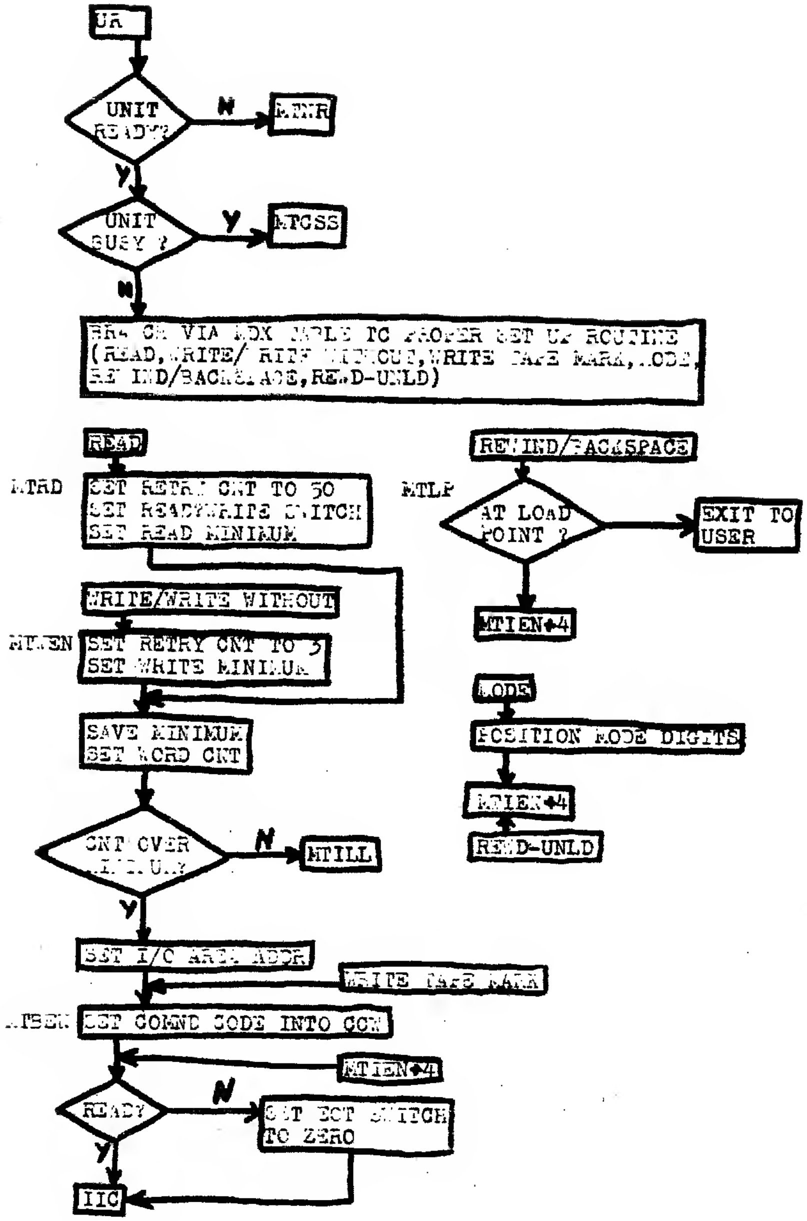
```
// FOR
*LISTALL   7-29.
*NAME TAPEM
*IOCS(CARD,1132 PRINTER)
      DIMENSION X(20)
      DO 5 K=1,9
      K=K+1
      READ(2,1)(X(I),I=1,18)
5     CALL MAGTA(2,0,36,X)
1     FORMAT(18A4)
      CALL MAGTA (5,0)
      CALL MAGTA (5,0)
      CALL MAGTA (3,0)
      DO 10 K=1,11
      K=K+1
      CALL MAGTA(0,0,36,X)
10    CALL MAGTA(2,1,36,X)
      CALL MAGTA (5,1)
      CALL MAGTA (5,1)
      CALL MAGTA (3,1)
      DO 15 K=1,9
      K=K+1
      CALL MAGTA(0,1,36,X)
15    WRITE(3,1)(X(I),I=1,18)
      CALL MAGTA(0,1,36,X)
      CALL MAGTA(0,1,36,X)
      CALL EXIT
      END

VARIABLE ALLOCATIONS
 X     =0026  K     =0028  I     =002A

STATEMENT ALLOCATIONS
 1     =0037  5     =006D  10    =0097  15    =00C1

FEATURES SUPPORTED
 IOCS

CALLED SUBPROGRAMS
 MAGTA    FLD      FSTO     SRED     SWRT     SCOMP    SFIO     SIOFX    SUBSC    CARDZ

INTEGER CONSTANTS
     1=002E       9=002F       2=0030       18=0031       0=0032       36=0033       5=

CORE REQUIREMENTS FOR TAPEM
 COMMON        0  VARIABLES      46  PROGRAM     192

END OF COMPILATION
```

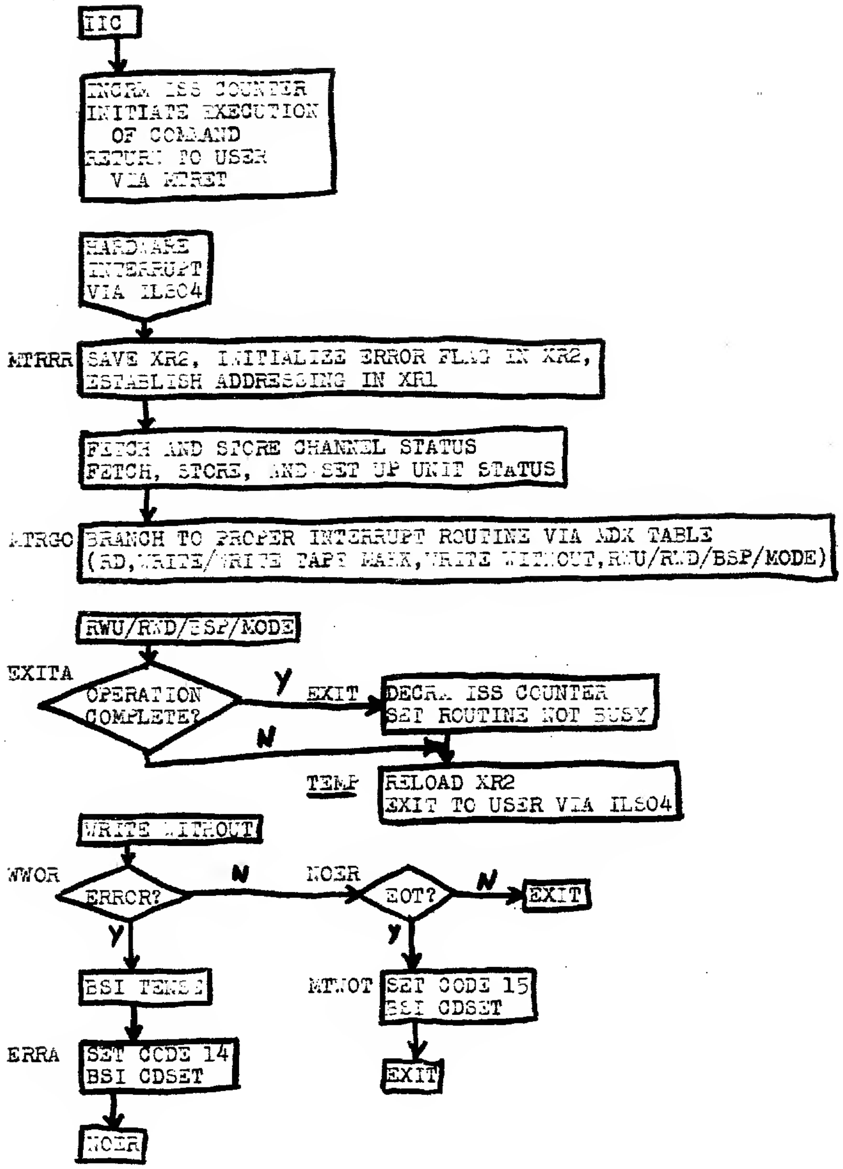
```
// XEQ TAPEM 7-29A.
```

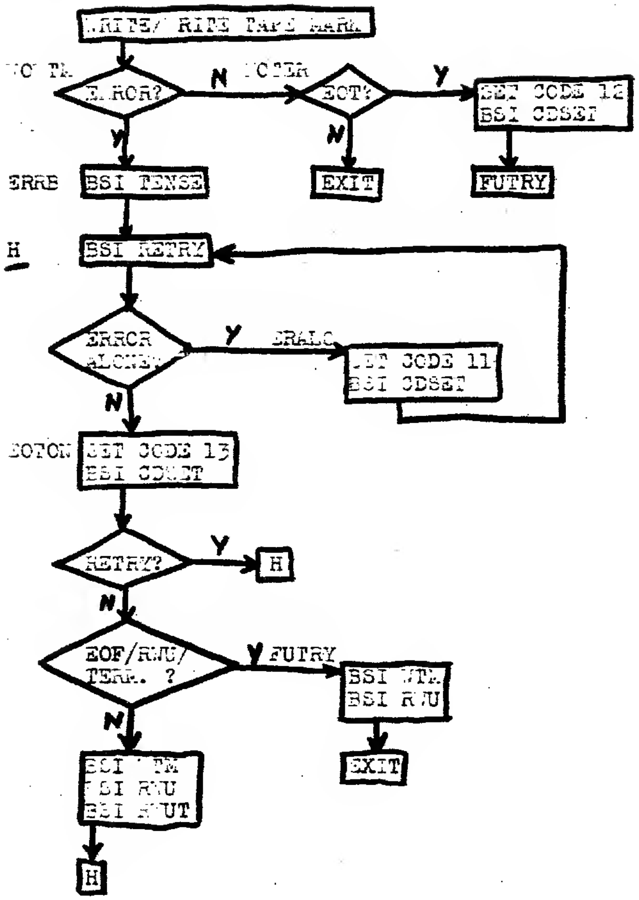
```
THIS PROGRAM TESTS THE MAGNETIC TAPE SUPPORT FOR ASSEMBLER PROGRAMS ON
THE IBM 1130 SYSTEM.  THE TEST CONSISTS OF READING 72 COLUMNS FROM
EACH OF FIVE DATA CARDS, WRITING THE CONTENTS OF EACH CARD ONTO TAPE
UNIT 0, TRANSFERING THE FIVE RECORDS FROM TAPE UNIT 0 TO TAPE UNIT 1,
AND FINALLY, READING THE RECORDS FROM TAPE UNIT 1 AND PRINTING THEM.
```

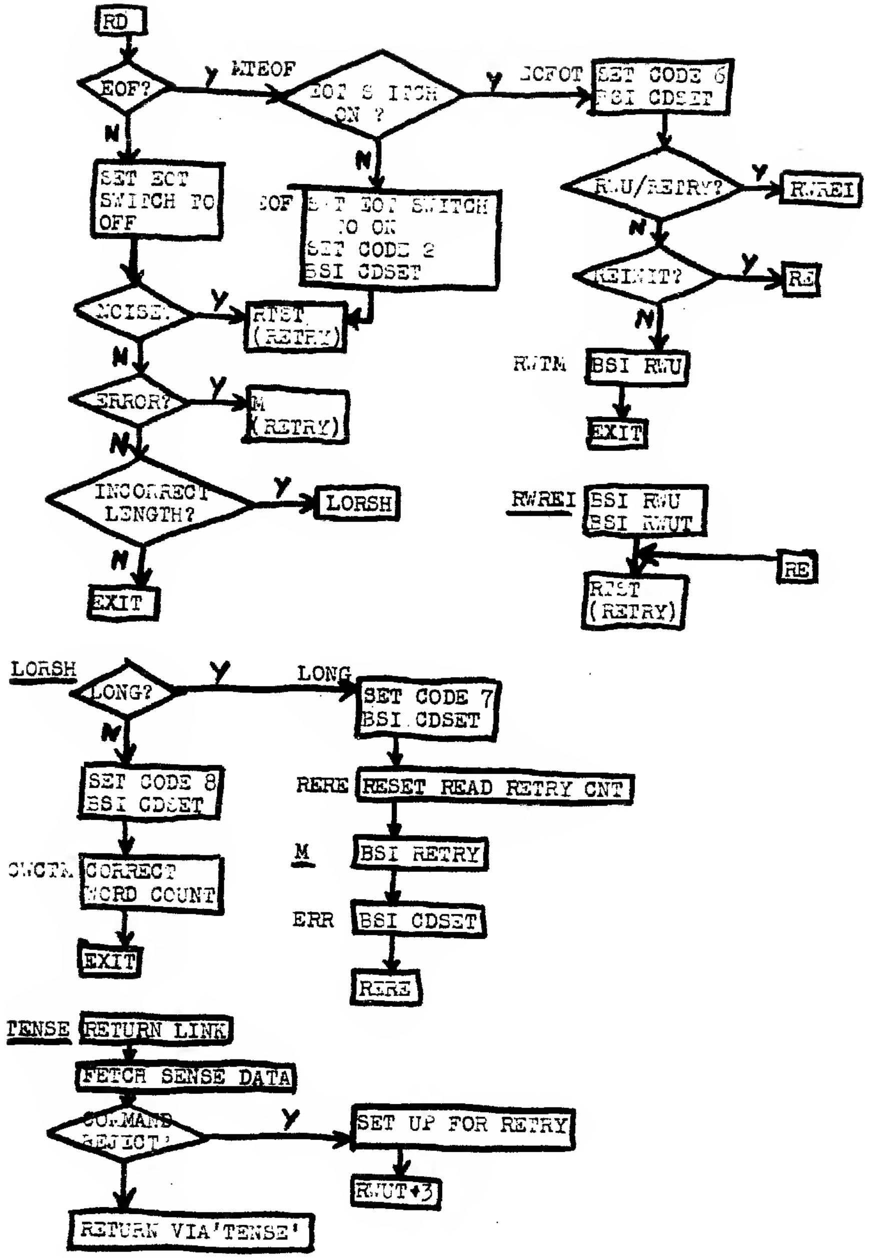
## 7-31. MAGT

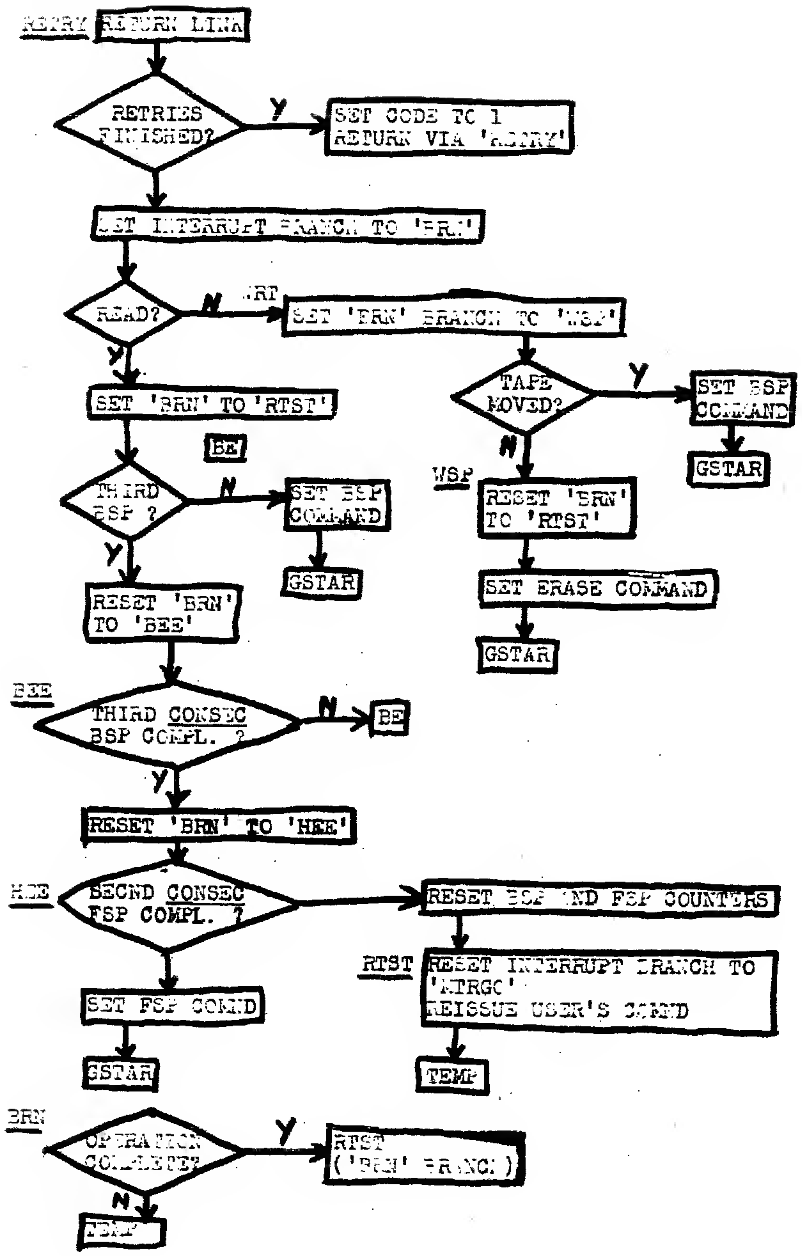

LIBF CALL
MAGT
LOAD ADDR OF PARAMETERS FROM T.V. SAVE XR2 AND XR1 AND ESTABLISH ADDRESSING IN XR1
STORE AC,EXT, AND STATUS
ROUTINE BUSY?
Y
N
FUNCTION 'TEST' ?
Y
N
SET RETURN VIA LIBF+2
FUNCTION 'TEST' ?
Y
SET RETURN VIA LIBF+3
N
RESTORE AC AND EXT
MTRET
RESTORE XR1, XR2,STATUS
RETURN TO USER
SET ROUTINE BUSY
MTNR
SET UP 'NOT READY' OR COMMAND REJECT (4X00) CODE
FUNCTION LEGAL ?
N
MTILL
SET UP ILLEGAL CODE (4001)
Y
SET UP TO RETRY CALL. EXIT TO LOC. 41
SET UP IOCC AND SEI ROUTINE ENTRANCE ADDRESSES
MTCSS
FETCH SENSE DATA
WAFET
TCU BUSY?
Y
N
UNIT EXISTS ?
N
MTILL
Y
UR

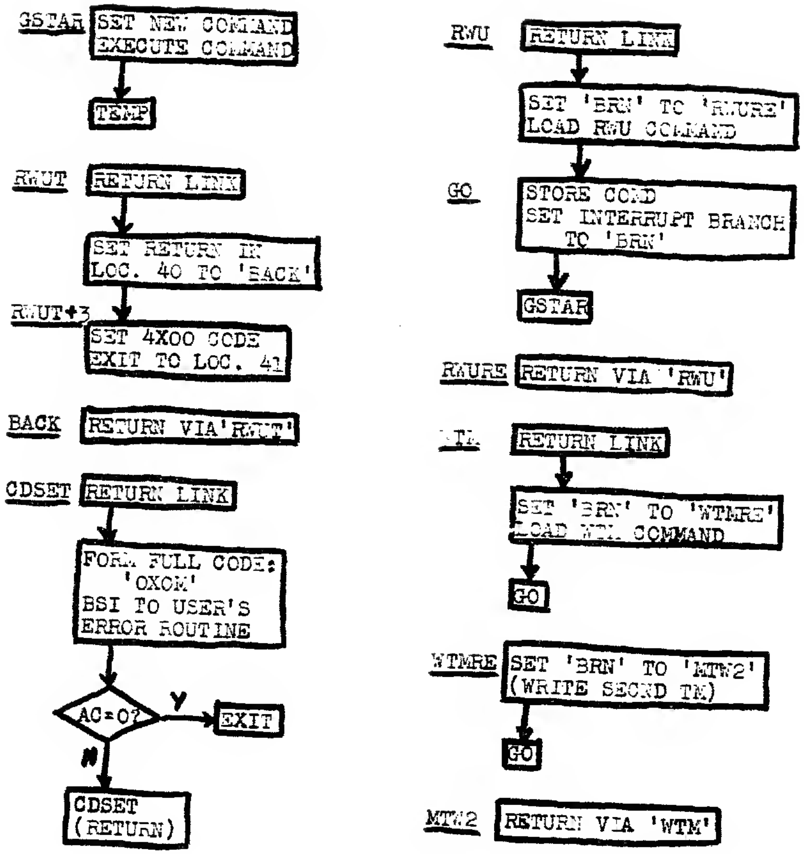

UR
UNIT READY?
N
MTNR
Y
UNIT BUSY ?
Y
MTCSS
N
BRANCH VIA NDX TABLE TO PROPER SET UP ROUTINE (READ,WRITE/WRITE WITHOUT,WRITE TAPE MARK,MODE, REWIND/BACKSPACE,REWD-UNLD)
READ
MTRD
SET RETRY CNT TO 50 SET READ/WRITE SWITCH SET READ MINIMUM
WRITE/WRITE WITHOUT
MTWEN
SET RETRY CNT TO 5 SET WRITE MINIMUM
SAVE MINIMUM SET WORD CNT
CNT OVER MINIMUM
N
MTILL
Y
SET I/O AREA ADDR
WRITE TAPE MARK
MTBEN
SET COMND CODE INTO CCW
MTIEN+4
READY
N
SET ECT SWITCH TO ZERO
Y
IIC
REWIND/BACKSPACE
MTLP
AT LOAD POINT ?
EXIT TO USER
MTIEN+4
MODE
POSITION MODE DIGITS
MTIEN+4
REWD-UNLD


IIC
INCRM ISS COUNTER
INITIATE EXECUTION
OF COMMAND
RETURN TO USER
VIA MTRET
HARDWARE
INTERRUPT
VIA ILS04
MTRRR
SAVE XR2, INITIALIZE ERROR FLAG IN XR2,
ESTABLISH ADDRESSING IN XR1
FETCH AND STORE CHANNEL STATUS
FETCH, STORE, AND SET UP UNIT STATUS
MTRGO
BRANCH TO PROPER INTERRUPT ROUTINE VIA MDX TABLE
(RD,WRITE/WRITE TAPE MARK,WRITE WITHOUT,RWU/RWD/BSP/MODE)
RWU/RWD/BSP/MODE
EXITA
OPERATION
COMPLETE?
Y
EXIT
DECRM ISS COUNTER
SET ROUTINE NOT BUSY
N
TEMP
RELOAD XR2
EXIT TO USER VIA ILS04
WRITE WITHOUT
WWOR
ERROR?
N
NOER
EOT?
N
EXIT
Y
BSI TENSE
Y
MTWOT
SET CODE 15
BSI CDSET
EXIT
ERRA
SET CODE 14
BSI CDSET
NOER

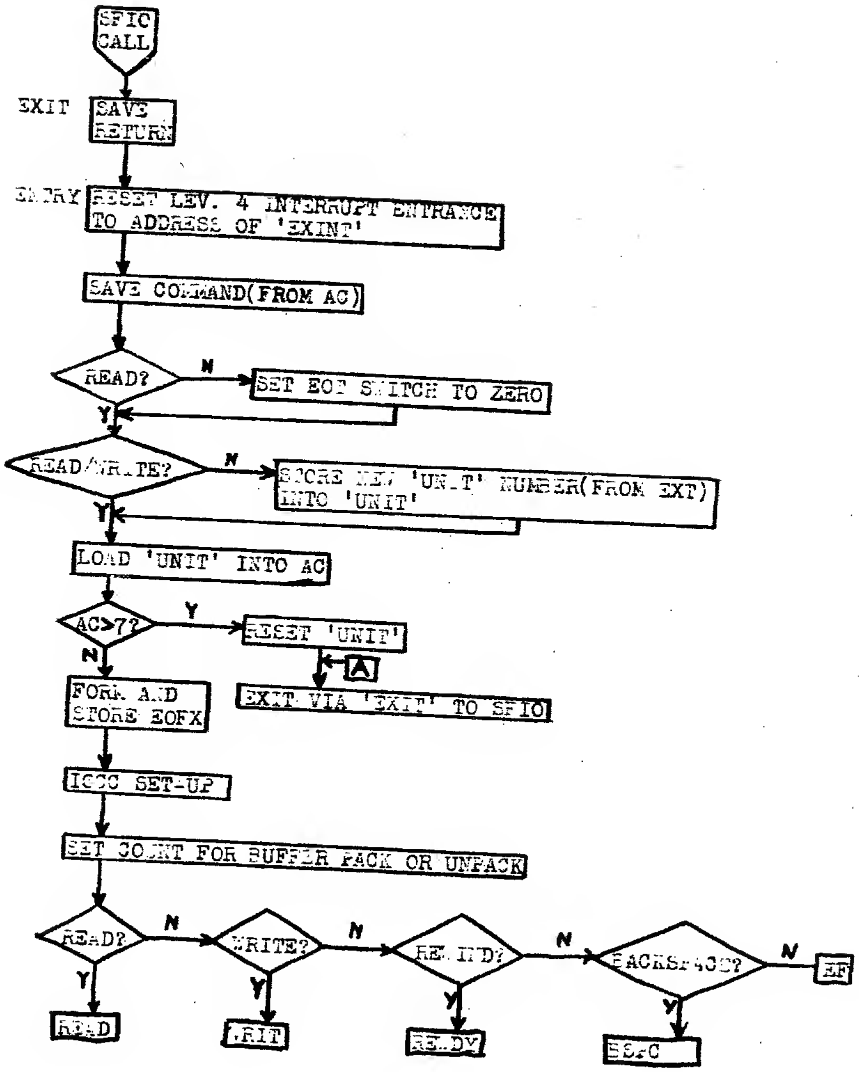

WRITE/WRITE TAPE MARK
WC TM
ERROR?
N
NOTER
EOT?
Y
SET CODE 12
BSI CDSET
Y
N
ERRB
BSI TENSE
EXIT
FUTRY
H
BSI RETRY
ERROR
ALONE?
Y
ERALO
SET CODE 11
BSI CDSET
N
EOTON
SET CODE 13
BSI CDSET
RETRY?
Y
H
N
EOF/RWU/
TERM. ?
Y
FUTRY
BSI WTM
BSI RWU
EXIT
N
BSI WTM
BSI RWU
BSI MTUT
H

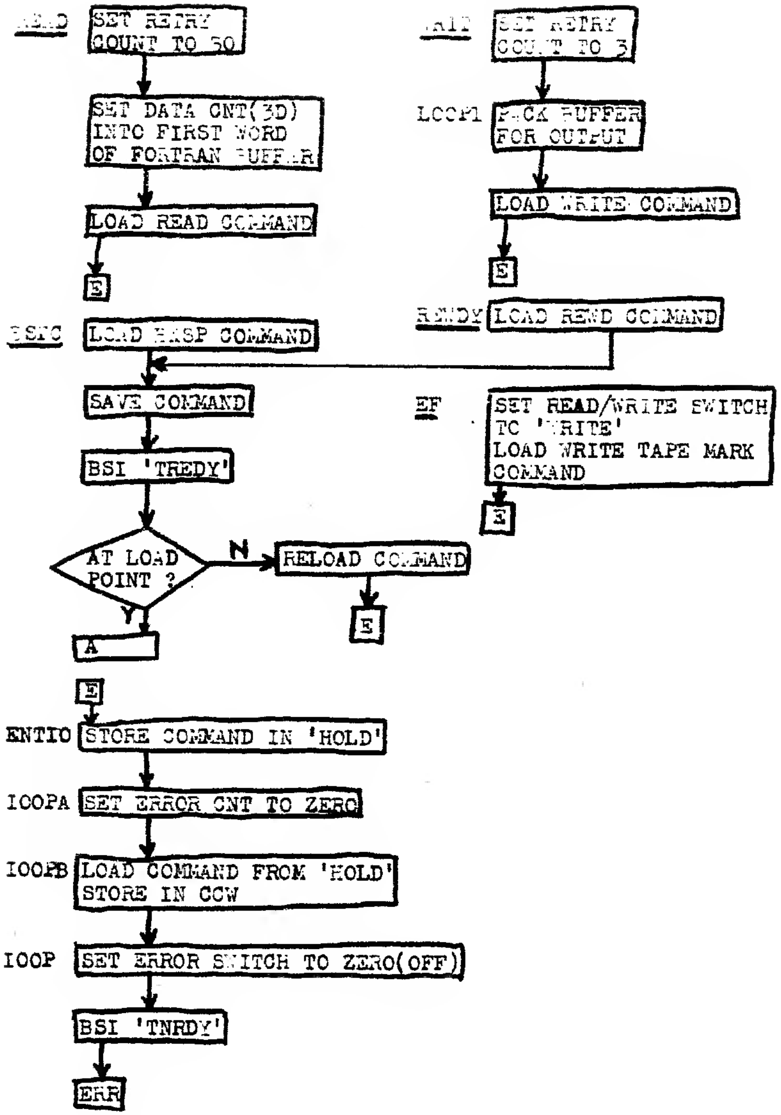

RD
EOF?
Y
MTEOF
EOT SWITCH ON ?
Y
EOFOT
SET CODE 6
BSI CDSET
N
SET EOT SWITCH TO OFF
EOF
SET EOT SWITCH TO ON
SET CODE 2
BSI CDSET
RWU/RETRY?
Y
RWREI
N
REINIT?
Y
RE
NOISE?
Y
RTST (RETRY)
N
ERROR?
Y
M (RETRY)
N
RWTM
BSI RWU
EXIT
INCORRECT LENGTH?
Y
LORSH
N
EXIT
RWREI
BSI RWU
BSI RWUT
RE
RTST (RETRY)
LORSH
LONG?
Y
LONG
SET CODE 7
BSI CDSET
N
SET CODE 8
BSI CDSET
RERE
RESET READ RETRY CNT
CWCTR
CORRECT WORD COUNT
M
BSI RETRY
EXIT
ERR
BSI CDSET
RERE
TENSE
RETURN LINK
FETCH SENSE DATA
COMMAND REJECT?
Y
SET UP FOR RETRY
RWUT+3
RETURN VIA 'TENSE'

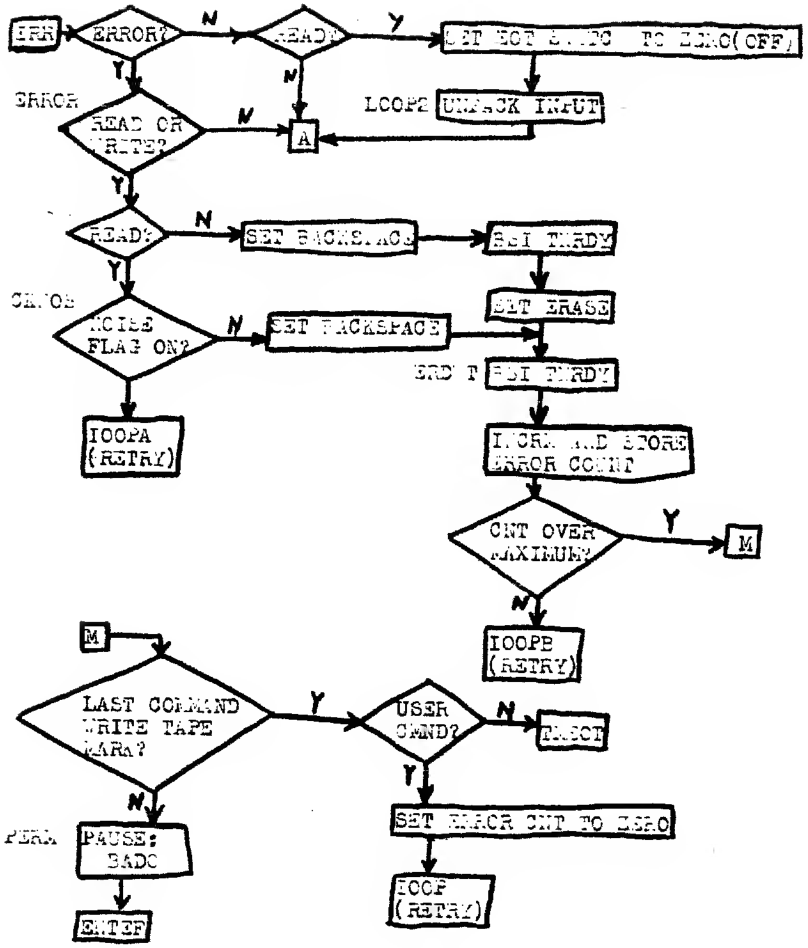

RETRY
RETURN LINK
RETRIES FINISHED?
Y
SET CODE TO 1
RETURN VIA 'RETRY'
SET INTERRUPT BRANCH TO 'BRN'
READ?
N
WRT
SET 'BRN' BRANCH TO 'WSP'
Y
SET 'BRN' TO 'RTST'
TAPE MOVED?
Y
SET BSP COMMAND
GSTAR
BE
THIRD BSP ?
N
SET BSP COMMAND
WSP
N
RESET 'BRN' TO 'RTST'
Y
GSTAR
RESET 'BRN' TO 'BEE'
SET ERASE COMMAND
GSTAR
BEE
THIRD CONSEC BSP COMPL. ?
N
BE
Y
RESET 'BRN' TO 'HEE'
HEE
SECND CONSEC FSP COMPL. ?
RESET BSP AND FSP COUNTERS
RTST
RESET INTERRUPT BRANCH TO 'WTRGO'
REISSUE USER'S COMND
SET FSP COMND
GSTAR
TEMP
BRN
OPERATION COMPLETE?
Y
RTST ('BRN' BRANCH)
N
TEMP

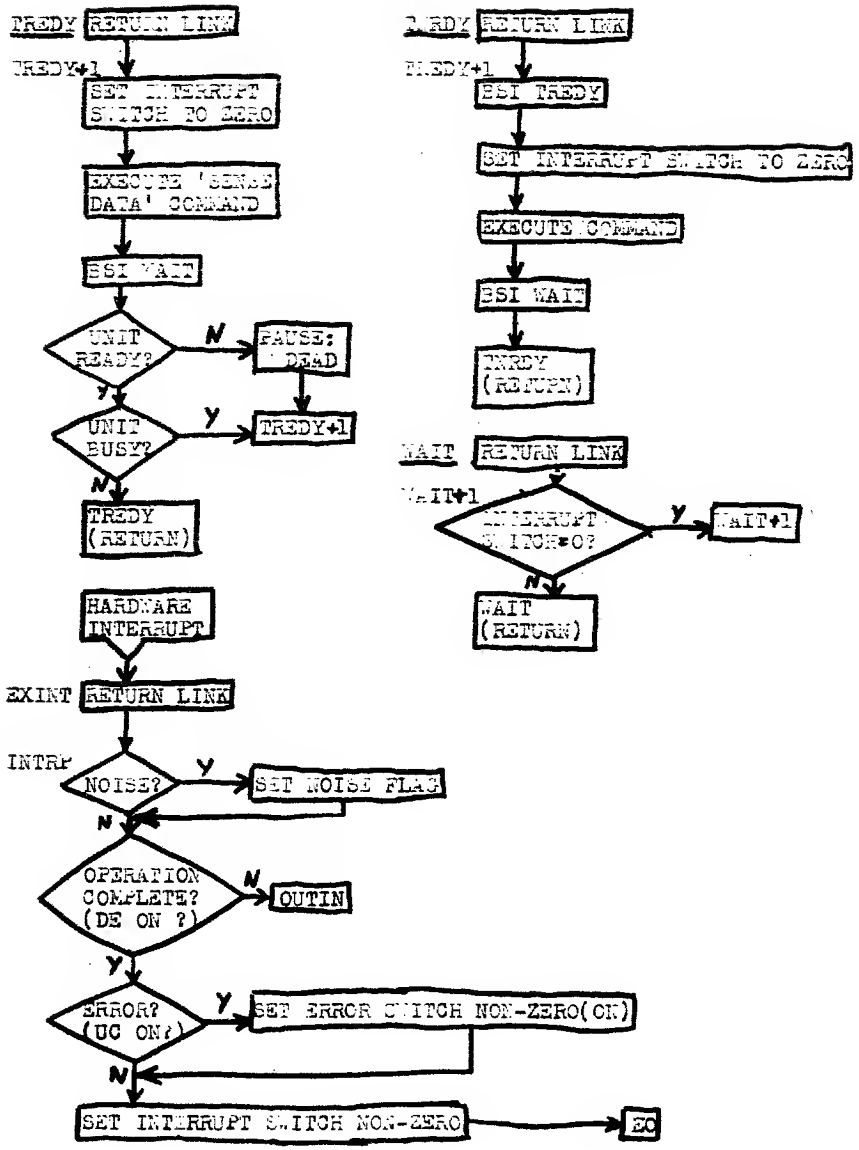

GSTAR
SET NEW COMMAND EXECUTE COMMAND
TEMP
RWUT
RETURN LINK
SET RETURN IN LOC. 40 TO 'BACK'
RWUT+3
SET 4X00 CODE EXIT TO LOC. 41
BACK
RETURN VIA 'RWUT'
CDSET
RETURN LINK
FORM FULL CODE: 'OXOM' BSI TO USER'S ERROR ROUTINE
AC=0?
Y
EXIT
N
CDSET (RETURN)
RWU
RETURN LINK
SET 'BRN' TO 'RWURE' LOAD RWU COMMAND
GO
STORE COMD SET INTERRUPT BRANCH TO 'BRN'
GSTAR
RWURE
RETURN VIA 'RWU'
WTM
RETURN LINK
SET 'BRN' TO 'WTMRE' LOAD WTM COMMAND
GO
WTMRE
SET 'BRN' TO 'MTW2' (WRITE SECND TM)
GO
MTW2
RETURN VIA 'WTM'

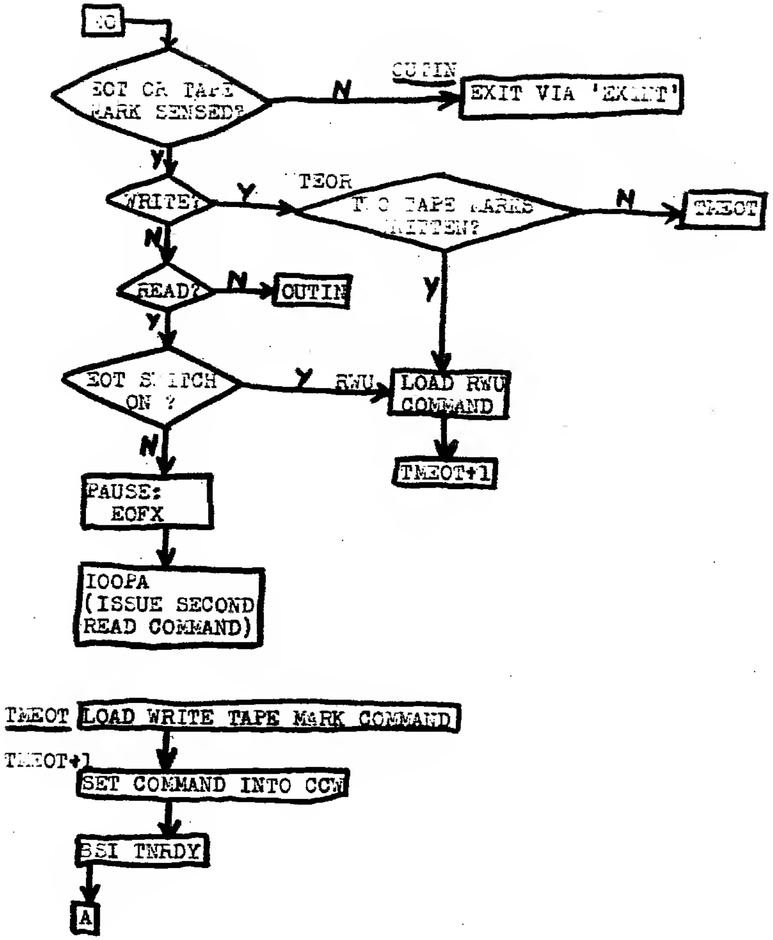
7-32. MAGTZ


SFIO CALL
EXIT
SAVE RETURN
ENTRY
RESET LEV. 4 INTERRUPT ENTRANCE TO ADDRESS OF 'EXINT'
SAVE COMMAND(FROM AC)
READ?
N
SET EOF SWITCH TO ZERO
Y
READ/WRITE?
N
STORE NEW 'UNIT' NUMBER(FROM EXT) INTO 'UNIT'
Y
LOAD 'UNIT' INTO AC
AC>7?
Y
RESET 'UNIT'
A
EXIT VIA 'EXIT' TO SFIO
N
FORM AND STORE EOFX
IOCC SET-UP
SET COUNT FOR BUFFER PACK OR UNPACK
READ?
N
WRITE?
N
REWIND?
N
BACKSPACE?
N
EF
Y
READ
Y
WRIT
Y
REDY
Y
BSPC

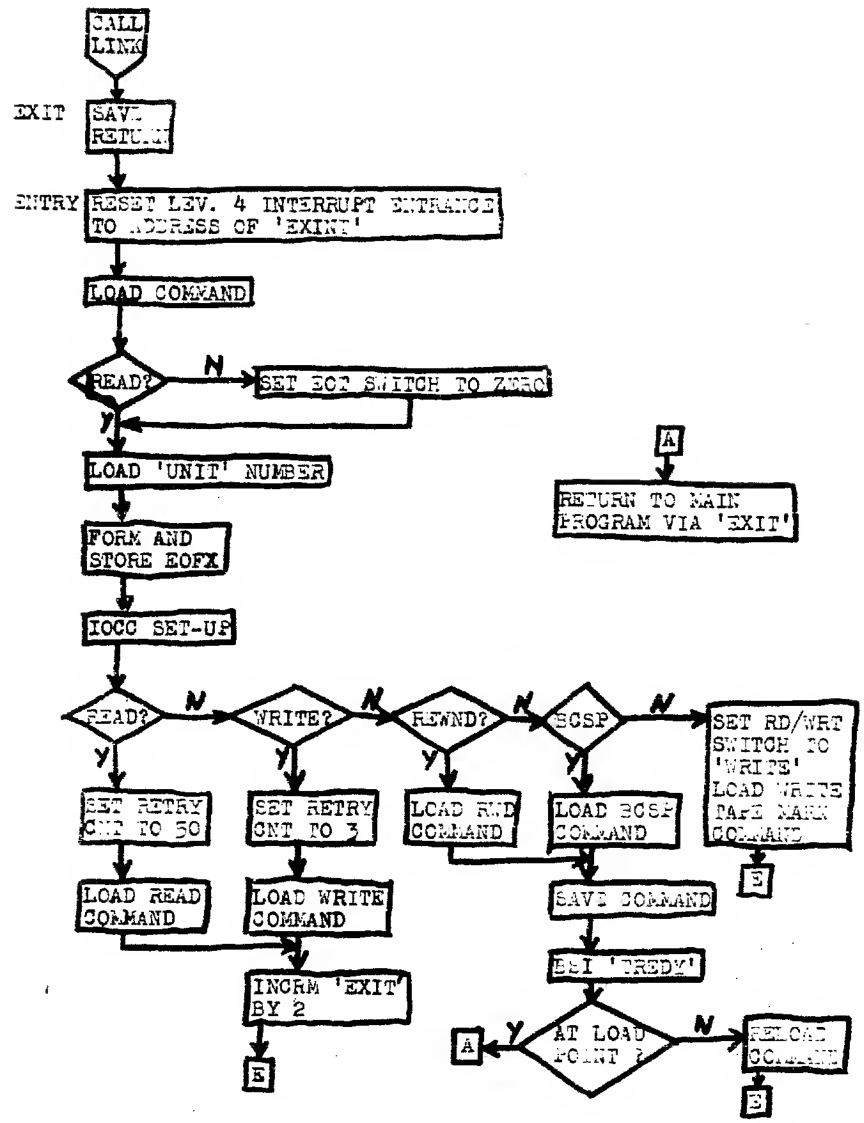

READ
SET RETRY COUNT TO 30
SET DATA CNT(3D) INTO FIRST WORD OF FORTRAN BUFFER
LOAD READ COMMAND
E
BSTC
LOAD BASP COMMAND
SAVE COMMAND
BSI 'TREDY'
AT LOAD POINT ?
N
Y
RELOAD COMMAND
E
A
WAIT
SET RETRY COUNT TO 3
LOOP1
PACK BUFFER FOR OUTPUT
LOAD WRITE COMMAND
E
RENDY
LOAD REND COMMAND
EF
SET READ/WRITE SWITCH TO 'WRITE'
LOAD WRITE TAPE MARK COMMAND
E
E
ENTIO
STORE COMMAND IN 'HOLD'
IOOPA
SET ERROR CNT TO ZERO
IOOPB
LOAD COMMAND FROM 'HOLD' STORE IN CCW
IOOP
SET ERROR SWITCH TO ZERO(OFF)
BSI 'TNRDY'
ERR

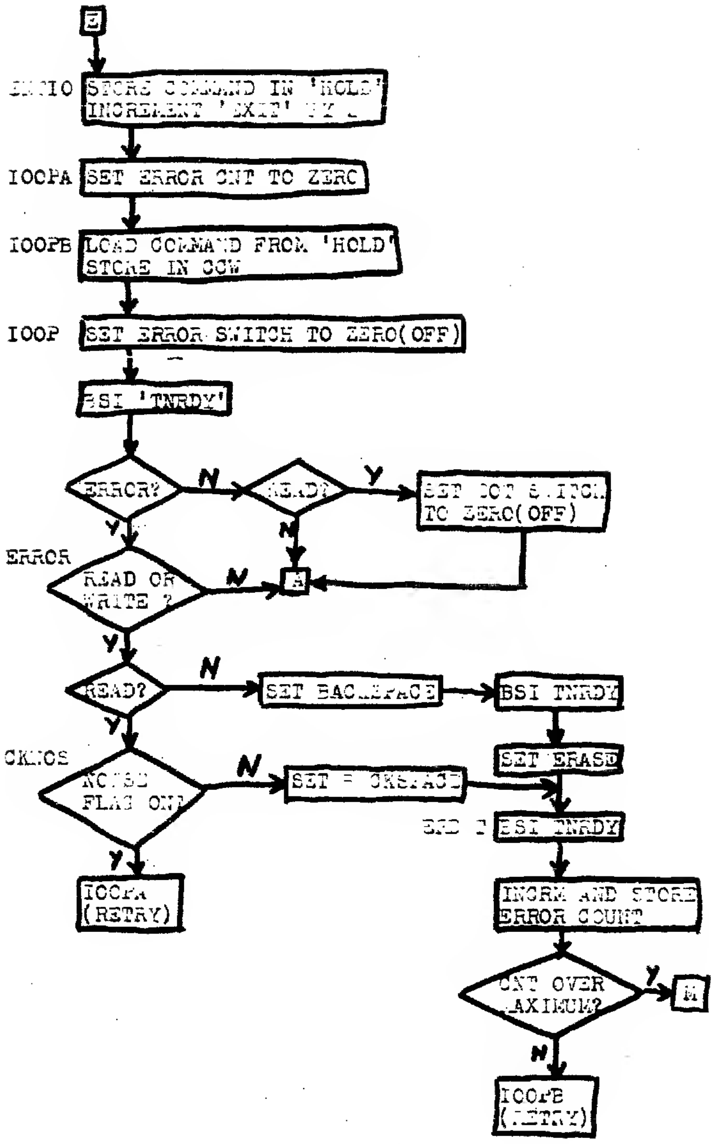

ERR
ERROR?
N
Y
READ?
Y
N
SET EOT & EOC TO ZERO(OFF)
LOOP2
UNPACK INPUT
ERROR
READ OR WRITE?
N
A
Y
READ?
N
SET BACKSPACE
BSI TNRDY
Y
SET ERASE
CHNOE
NOISE FLAG ON?
N
SET BACKSPACE
ERDT
BSI TNRDY
IOOPA (RETRY)
INCR AND STORE ERROR COUNT
CNT OVER MAXIMUM?
Y
M
N
IOOPB (RETRY)
M
LAST COMMAND WRITE TAPE MARK?
Y
USER CMND?
N
TLSCT
Y
N
PERR
PAUSE: BADC
SET ERROR CNT TO ZERO
ENTEF
IOOP (RETRY)

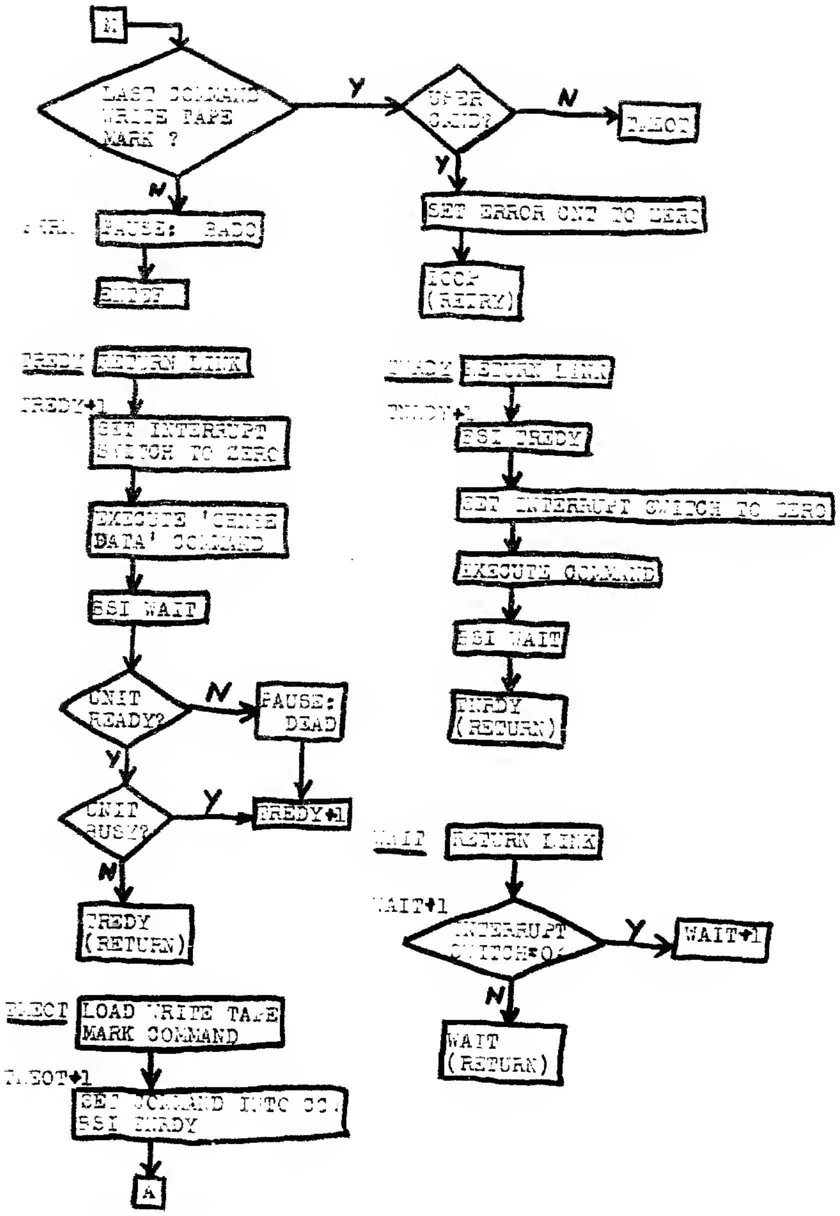

TREDY
RETURN LINK
TREDY+1
SET INTERRUPT SWITCH TO ZERO
EXECUTE 'SENSE DATA' COMMAND
BSI WAIT
UNIT READY?
N
PAUSE: DEAD
Y
UNIT BUSY?
Y
TREDY+1
N
TREDY (RETURN)
TNRDY
RETURN LINK
TNRDY+1
BSI TREDY
SET INTERRUPT SWITCH TO ZERO
EXECUTE COMMAND
BSI WAIT
TNRDY (RETURN)
WAIT
RETURN LINK
WAIT+1
INTERRUPT SWITCH=0?
Y
WAIT+1
N
WAIT (RETURN)
HARDWARE INTERRUPT
EXINT
RETURN LINK
INTRP
NOISE?
Y
SET NOISE FLAG
N
OPERATION COMPLETE? (DE ON ?)
N
OUTIN
Y
ERROR? (UC ON?)
Y
SET ERROR SWITCH NON-ZERO(ON)
N
SET INTERRUPT SWITCH NON-ZERO
EC


EC
EOT OR TAPE MARK SENSED?
N
OUTIN
EXIT VIA 'EXIT'
Y
WRITE?
Y
TEOR
TWO TAPE MARKS WRITTEN?
N
TMEOT
N
READ?
N
OUTIN
Y
Y
EOT SWITCH ON ?
Y
RWU
LOAD RWU COMMAND
TMEOT+1
N
PAUSE: EOFX
IOOPA (ISSUE SECOND READ COMMAND)
TMEOT
LOAD WRITE TAPE MARK COMMAND
TMEOT+1
SET COMMAND INTO CCW
BSI TNRDY
A

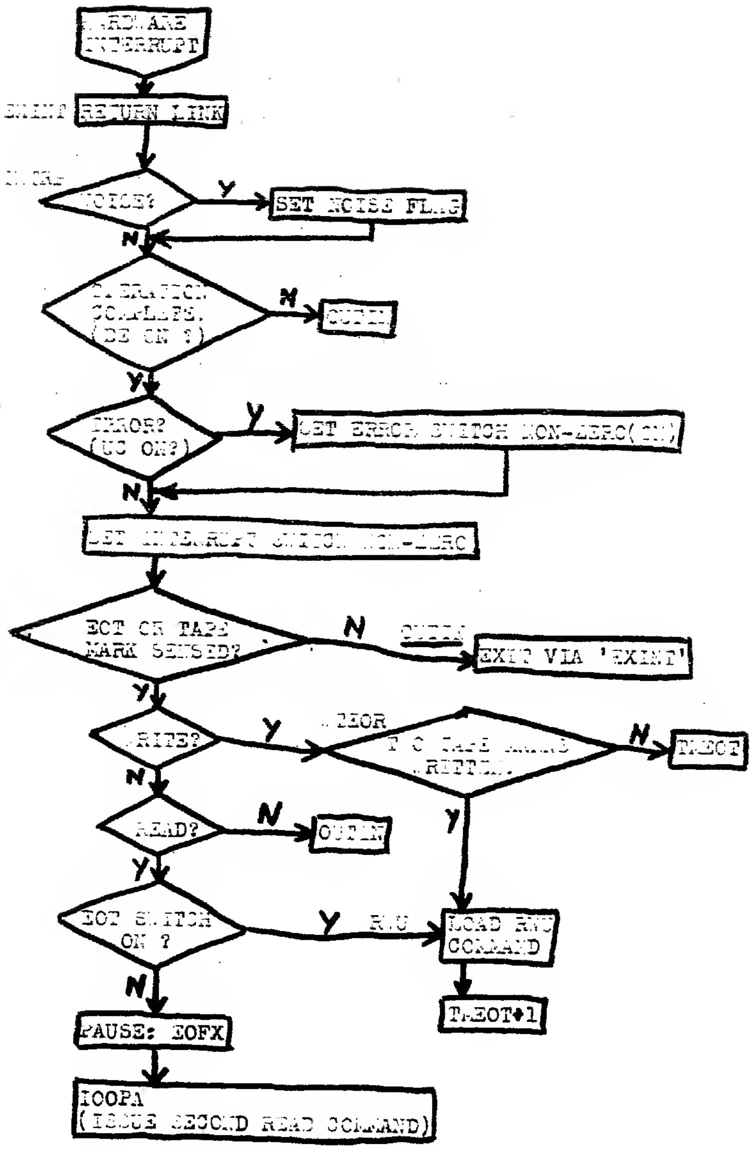
7-33. MAGTA


CALL LINK
EXIT
SAVE RETURN
ENTRY
RESET LEV. 4 INTERRUPT ENTRANCE TO ADDRESS OF 'EXINT'
LOAD COMMAND
READ?
N
SET EOT SWITCH TO ZERO
Y
LOAD 'UNIT' NUMBER
FORM AND STORE EOFX
IOCC SET-UP
READ?
N
WRITE?
N
REWND?
N
BCSP
N
SET RD/WRT SWITCH TO 'WRITE' LOAD WRITE TAPE MARK COMMAND
E
Y
SET RETRY CNT TO 50
LOAD READ COMMAND
Y
SET RETRY CNT TO 3
LOAD WRITE COMMAND
INCRM 'EXIT' BY 2
E
Y
LOAD RWD COMMAND
Y
LOAD BCSP COMMAND
SAVE COMMAND
BSI 'TREDY'
A
Y
AT LOAD POINT ?
N
RELOAD COMMAND
E
A
RETURN TO MAIN PROGRAM VIA 'EXIT'


E
ENTIO
STORE COMMAND IN 'HOLD' INCREMENT 'EXIT' BY 1
IOOPA
SET ERROR CNT TO ZERO
IOOPB
LOAD COMMAND FROM 'HOLD' STORE IN CCW
IOOP
SET ERROR SWITCH TO ZERO(OFF)
BSI 'TNRDY'
ERROR?
N
READ?
Y
SET EOT SWITCH TO ZERO(OFF)
N
Y
ERROR
READ OR WRITE ?
N
A
Y
READ?
N
SET BACKSPACE
BSI TNRDY
SET ERASE
Y
CKNOS
NOISE FLAG ON?
N
SET BACKSPACE
ERD T
BSI TNRDY
Y
IOOPA (RETRY)
INCRM AND STORE ERROR COUNT
CNT OVER MAXIMUM?
Y
M
N
IOOPB (RETRY)



M
LAST COMMAND WRITE TAPE MARK ?
Y
USER CALLED?
N
TMEOT
Y
SET ERROR CNT TO ZERO
LOOP (RETRY)
N
PAUSE: BADC
EMPTY
TREDY
RETURN LINK
TREDY+1
SET INTERRUPT SWITCH TO ZERO
EXECUTE 'SENSE DATA' COMMAND
ESI WAIT
UNIT READY?
N
PAUSE: DEAD
Y
UNIT BUSY?
Y
TREDY+1
N
TREDY (RETURN)
TMEOT
LOAD WRITE TAPE MARK COMMAND
TMEOT+1
SET COMMAND INTO CC. ESI THRDY
A
THRDY
RETURN LINK
THRDY+1
ESI TREDY
SET INTERRUPT SWITCH TO ZERO
EXECUTE COMMAND
ESI WAIT
THRDY (RETURN)
WAIT
RETURN LINK
WAIT+1
INTERRUPT SWITCH=0?
Y
WAIT+1
N
WAIT (RETURN)



HARDWARE INTERRUPT
EXINT
RETURN LINK
INTRP
NOISE?
Y
SET NOISE FLAG
N
OPERATION COMPLETE. (DE ON ?)
N
OUTIN
Y
ERROR? (UC ON?)
Y
SET ERROR SWITCH NON-ZERO(ON)
N
SET INTERRUPT SWITCH NON-ZERO
EOT OR TAPE MARK SENSED?
N
OUTIN
EXIT VIA 'EXINT'
Y
WRITE?
Y
ERROR
TWO TAPE MARKS WRITTEN
N
TMEOT
N
READ?
N
OUTIN
Y
EOT SWITCH ON ?
Y
RWU
Y
LOAD RWU COMMAND
TMEOT+1
N
PAUSE: EOFX
LOOPA (ISSUE SECOND READ COMMAND)

## 7 - 34. ILS04, REWNZ, IOU, AND SFIO

Flowcharts of these routines have NOT been included since they are basically standard system subroutines.

ILS04

Standard level 4 interrupt routine except for changes(indicated by arrows, cf. 7 - 23.) needed to test for interrupts from the 2954 R.P.Q. Selector Channel.

REWNZ

Interface routine for Fortran and MAGTZ for BACKSPACE, END FILE, and REWIND commands (cf. 7 - 27.).

IOU

Converts logical unit numbers to physical unit numbers; is called by REWNZ ( cf. 7 - 26.).

SFIO

Main 1130 single device I/O Fortran routine with the test for an illegal device on a READ operation disabled. The original routine considered all odd numbered devices (e.g. console printer, printer, plotter) as illegal. However, since magnetic tape is number five, this method of testing the device number is clearly inadequate. The test should be re-written and the entire routine reassembled instead of just being disabled, but SFIO is a large routine and no source deck was readily available, so the test was dis abled by making a BSC L instruction into an unconditional branch: this required changing only one bit in the entire program and could be done easily with an object deck.

## APPENDIX A. ERRORS DETECTED BY MAGT SUBROUTINE*

| Error | Accumulator Contents(hex) |
|---|---|
| Write and Write Tape Mark | |
| *Error | 0 X 0 B |
| *End-Of-Tape | 0 X 0 C |
| *Error/EOT | 0 X 0 D |
| Write Without Retries | |
| *Error | 0 X 0 E |
| *End-Of-Tape | 0 X 0 F |
| Read | |
| *Error | 0 X 0 1 |
| *End-Of-File | 0 X 0 2 |
| *EOT | 0 X 0 6 |
| *Long Record | 0 X 0 7 |
| *Short Record | 0 X 0 8 |
| Device not ready or command reject | 4 X 0 0 |
| Illegal unit, functin, or word count | 4 0 0 1 |

*The errors marked with an asterisk cause a branch via the error parameter. These errors are detected during the processing of interrupts; as a consequence, the user's error routine is an interrupt routine, executed at priority level 4.

All other errors cause a branch to location 41. The address of the LIBF in error is in location 40.

X's correspond to the device identification digit in the related calling sequence.

## APPENDIX B. MAGT SUBROUTINE ACTION AFTER RETURN FROM USER*

| Error Code | Condition | Subr. Action |
|---|---|---|
| Write and Write Tape Mark | | |
| 0 X 0 B | If AC is 0 | Terminate |
| | Otherwise | Retry |
| 0 X 0 C | If AC is 0 | Terminate |
| | Otherwise | EOF/EOF/RWU/Term. |
| 0 X 0 D | If AC is 0 | Terminate |
| | If AC is negative | Retry |
| | If AC is odd/pos | EOF/EOF/RWU/Term. |
| | If AC is even/pos | EOF/EOF/RWU/Retry |
| Write Without Retries | | |
| 0 X 0 E | If AC is 0 | Terminate |
| | Otherwise | Check for EOT** |
| 0 X 0 F | In any case | Terminate |
| Read | | |
| 0 X 0 1 | If AC is 0 | Terminate |
| | Otherwise | Retry |
| 0 X 0 2 | If AC is 0 | Terminate |
| | Otherwise | Reinitiate |
| 0 X 0 6 | If AC is 0 | Terminate |
| | If AC is negative | RWU/Reinitiate |
| | If AC is odd/pos | Reinitiate |
| | If AC is even/pos | RWU/Terminate |
| 0 X 0 7 | If AC is 0 | Terminate |
| | Otherwise | Retry |
| 0 X 0 8 | If AC is 0 | Terminate |
| | Otherwise | Correct Count/Term. |

*For Rewind/Unload commands and RWU/Terminate recovery choices, the subroutine is set not busy, other tape commands on other units may be executed, and the unloaded unit may be reloaded at any time. For RWU/Retry and RWU/Reinitiate recovery choices, the subroutine remains busy and no other tape commands can be executed until the unloaded unit is reloaded and execution of the current recovery choice is completed. While waiting for the unit to be reloaded, the routine presents the error code for 'device not ready' (4X00) and maintains a wait state at location 41.

**If EOT, 0 X 0 F is indicated to the user's error routine; if not EOT, the operation is terminated.

## APPENDIX C. MAGTA AND MAGTZ ERRORS DETECTED AND USER ACTION

| Error/AC Code | User Action ⟶ | Subr. Action |
|---|---|---|
| Device not ready (D E A D) | Ready device, press program start | Current command retried |
| Non-correctable read, write, or end file error (B A D C) | Press program start | Current command terminated, but program execution continued at next command |
| Read | | |
| Tape mark sensed (E O F X) | Press program start | Current read instruction tried on next record |
| EOT condition satisfied | (NO action needed) | Tape unit rewound/unloaded; program execution continued at next command |
| Write or End File | | |
| EOT condition satisfied | (NO action needed) | Two tape marks are written on tape; tape unit rewound/unloaded; program execution continues at next command |